caBench-to-Bedside v3.1

Administration Manual

Document Change History

Version Number	Date	Contributor	Description
V1.0	June 25, 2008	Washington University Persistent Systems	Reviewed
V2.1	December 4, 2008	Washington University Persistent Systems	Reviewed
V2.1	December 23,2008	Washington University Persistent Systems	Reviewed by Documentation and Training workspace
V3.0	June 26, 2009	Washington University Persistent Systems	Reviewed
V3.0	July 25, 2009	Washington University Persistent Systems	Addressed comments by Documentation and Training workspace
V3.1	September 12, 2009	Washington University Persistent Systems	Reviewed
V3.1	September 16, 2009	Washington University Persistent Systems	Reviewed by Documentation and Training workspace

caBIG®
cancer Biomedical
Informatics Grid®

September 23, 2009

Development Teams		
Development	***End User Manual***	***Program Management***
Chandrakant Talele[1]	Chandrakant Talele[1]	Dr. Rakesh Nagarajan[2]
Chetan Patil[1]	Madhumita Shrikhande[1]	Srikanth Adiga[1, 2]
Chetan Pundhir[1]	Pooja Arora[1]	
Deepak Shingan[1]	Srikanth Adiga[1,2]	
Gaurav Mehta[1]	Mukesh Sharma[2]	
Pallavi Mistry[1]		
Rajesh Vyas[1]		
[1]Persistent Systems Limited	[2]Washington University	

Contacts and Support	
Training contact	cab2bsupport@bmi.wustl.edu
Customer support contact	cab2bsupport@bmi.wustl.edu

Copyright and License Information

Copyright Notice. Copyright 2005 "School of Medicine, Washington University in St. Louis" ("caBIG® Participant"). cancer Bench-to-Bedside was created with NCI funding and is part of the caBIG® initiative. The software subject to this notice and license includes both human readable source code form and machine readable, binary, object code form (the "caBIG® Software").

This caBIG® Software License (the "License") is between caBIG® Participant and You. "You (or "Your") shall mean a person or an entity, and all other entities that control, are controlled by, or are under common control with the entity. "Control" for purposes of this definition means (i) the direct or indirect power to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

License. Provided that You agree to the conditions described below, caBIG® Participant grants You a non-exclusive, worldwide, perpetual, fully-paid-up, no-charge, irrevocable, transferable and royalty-free right and license in its rights in the caBIG® Software, including any copyright or patent rights therein that may be infringed by the making, using, selling, offering for sale, or importing of caBIG® Software, to (i) use, install, access, operate, execute, reproduce, copy, modify, translate, market, publicly display, publicly perform, and prepare derivative works of the caBIG® Software; (ii) make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of caBIG® Software (or portions thereof); (iii) distribute and have distributed to and by third parties the caBIG® Software and any modifications and derivative works thereof; and (iv) sublicense the foregoing rights set out in (i), (ii) and (iii) to third parties, including the right to license such rights to further third parties. For sake of clarity, and not by way of limitation, caBIG® Participant shall have no right of accounting or right of payment from You or Your sublicensees for the rights granted under this License. This License is granted at no charge to You. Your downloading, copying, modifying, displaying, distributing or use of caBIG® Software constitutes acceptance of all of the terms and conditions of this Agreement. If you do not agree to such terms and conditions, you have no right to download, copy, modify, display, distribute or use the caBIG® Software.

1. Your redistributions of the source code for the caBIG® Software must retain the above copyright notice, this list of conditions and the disclaimer and limitation of liability of Article 6 below. Your redistributions in object code form must reproduce the above copyright notice, this list of conditions and the disclaimer of Article 6 in the documentation and/or other materials provided with the distribution, if any.

2. Your end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by School of Medicine, Washington University in St. Louis." If You do not include such end-user documentation, You shall include this acknowledgment in the caBIG® Software itself, wherever such third-party acknowledgments normally appear.

3. You may not use the names "School of Medicine, Washington University in St. Louis", "The National Cancer Institute", "NCI", "Cancer Bioinformatics Grid" or "caBIG®" to endorse or promote products derived from this caBIG® Software. This License does not authorize You to use any trademarks, service marks, trade names,

logos or product names of either caBIG® Participant, NCI or caBIG®, except as required to comply with the terms of this License.

4. For sake of clarity, and not by way of limitation, You may incorporate this caBIG® Software into Your proprietary programs and into any third party proprietary programs. However, if You incorporate the caBIG® Software into third party proprietary programs, You agree that You are solely responsible for obtaining any permission from such third parties required to incorporate the caBIG® Software into such third party proprietary programs and for informing Your sublicensees, including without limitation Your end-users, of their obligation to secure any required permissions from such third parties before incorporating the caBIG® Software into such third party proprietary software programs. In the event that You fail to obtain such permissions, You agree to indemnify caBIG® Participant for any claims against caBIG® Participant by such third parties, except to the extent prohibited by law, resulting from Your failure to obtain such permissions.

5. For sake of clarity, and not by way of limitation, You may add Your own copyright statement to Your modifications and to the derivative works, and You may provide additional or different license terms and conditions in Your sublicenses of modifications of the caBIG® Software, or any derivative works of the caBIG® Software as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

6. THIS caBIG® SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES (INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE) ARE DISCLAIMED. IN NO EVENT SHALL THE SCHOOL OF MEDICINE, WASHINGTON UNIVERSITY IN ST. LOUIS OR ITS AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS caBIG® SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Contents

About This Guide

This section introduces you to the cancer Bench-To-Bedside (caB2B) Administrative Module. Topics in this section include:

- *Purpose*
- *Typical User*
- *Topics Covered*
- *Text Conventions Used*

Purpose

This manual provides the details of how to use **ca**ncer **B**ench-To-**B**edside (caB2B) Administrative Module to set up, configure, and manage the Client Application and Web Application.

All the chapters in this administration manual begin with an introductory statement followed by a list of topics included in that chapter.

Typical User

This guide is designed for the administrators of caBench-To-Bedside (caB2B) application, who will configure and manage the caB2B applications. Administrator should have knowledge of bioinformatics that will be helpful to understand various caBIG® applications and their models. Administrator should have an understanding of classes and attributes present in a domain (UML) model. Based on the knowledge, administrator will create useful queries for the novice end users who do not have knowledge of UML model, caGrid etc.

Administrator should have intermediate knowledge of computers. For activities like creating multi-model category, knowledge of Extensible Markup Language (XML) and basic knowledge of executing commands is desired.

Topics Covered

- "*Chapter 1: Introduction*" gives brief introduction to application
- *"Chapter 2: Load Models"* describes process of loading caDSR metadata
- *"Chapter 3: Define Model Groups"* describes creation of model groups
- *"Chapter 4: Service Instances"* describes how to configure service instances
- *"Chapter 5: Create Category"* describes process of category creation
- *"Chapter 6: Curate Paths"* describes how paths are curated and their use
- *"Chapter 7: Inter Model Joins"* describes how to connect two applications
- *"Chapter 8: Multi-model Category"* describes process of multi-model category creation

Text Conventions Used

This section explains conventions used in this guide. The various typefaces represent interface components, keyboard shortcuts, toolbar buttons, dialog box options, and text that you type.

Convention	***Description***	***Example***
Bold	Highlights names of option buttons, check boxes, drop-down menus, menu commands, command buttons, or icons	Click **Search**.
URL	Indicates a Web address	http://domain.com
Text in SMALL CAPS	Indicates a keyboard shortcut	Press ENTER.
text in SMALL CAPS + text in SMALL CAPS	Indicates keys that are pressed simultaneously	Press SHIFT + CTRL
Italics	Highlights references to other documents, sections, figures, and tables	See *Figure 4.5*.
Italic boldface monospace type	Represents text that you type	In the **New Subset** text box, enter ***Proprietary Proteins***.
{ }	Surrounds replaceable items	Replace {last name, first name} with the Principal Investigator's name.
F.Y.I.	Indicates information of particular interest	F.Y.I. Administrator can create a new category.
CAUTION	Indicates that you should be particularly aware of the information provided	CAUTION You cannot modify path

Chapter 1 Introduction

This chapter explains the scope and function of the cancer Bench-To-Bedside (caB2B) Administrative Module. It explains how the caB2B Administrative Module helps in the configuring and setting up of the caB2B Client Application and the Web Application.

This chapter contains the following topics:

- *Introduction to Administrative Module*
- *Starting the caB2B Administrative Module*

Introduction to Administrative Module

cancer **B**ench-to-**B**edside (caB2B) is an application that leverages caBIG® tools in a user-friendly graphical user interface (GUI). caB2B is a tool that provides a platform for querying all caBIG tools. Before launching the caB2B Client Application and Web Application, the administrator should configure it for the users. S/he should load models from cancer Data Standards Repository (caDSR), create useful categories, curate the paths between the classes, create intermodel joins, and define multi-model categories. These customized categories and curated paths make the application easier for the end users.

The Administrative Module is a web-based application used to carry out the following tasks:

- **Load new models from the caDSR**

  Administrator load models present in the cancer Data Standards Repository (caDSR). Once the model is loaded, its metadata would be available for query.

- **Define defaults for services to be used**

- **Define Model groups**

  To avoid complexity of UML models like their specific names, different versions etc from end users, administrator defines model group using one or more models. Web Application users can access the data from the available models in model group.

- **Define the categories to be queried**

  Administrator defines categories by combining multiple classes from the domain model, to make the task of querying easier for the end-user

- **Curate path between categories**

  By default, there are several paths available between two categories. From these paths, the administrator can select only those paths that are useful for the end-user.

- **Define Inter model joins**

Administrator makes two models interoperable by defining a bridge between those using classes having common data element or manually specifying the joining attributes.

- **Define Multi-model Category**

  Administrator creates multi-model category using different applications that store same kind of data. This category helps in querying across multiple applications or different versions of the same applications.

Starting the caB2B Administrative Module

You or your technical staff can install the application following the instructions provided in the *caB2B Installation Manual*. After installing the application, you can access the URL mentioned in the installation manual and access the caB2B administrative module for configuring Client Application and Web Application. To know more about how these modules are connected to each other refer to the *caB2B Installation Manual.*

When you access the URL for the Administrative Module the system navigates to the login page of the Administrative Module, as shown below in Figure 1.1.

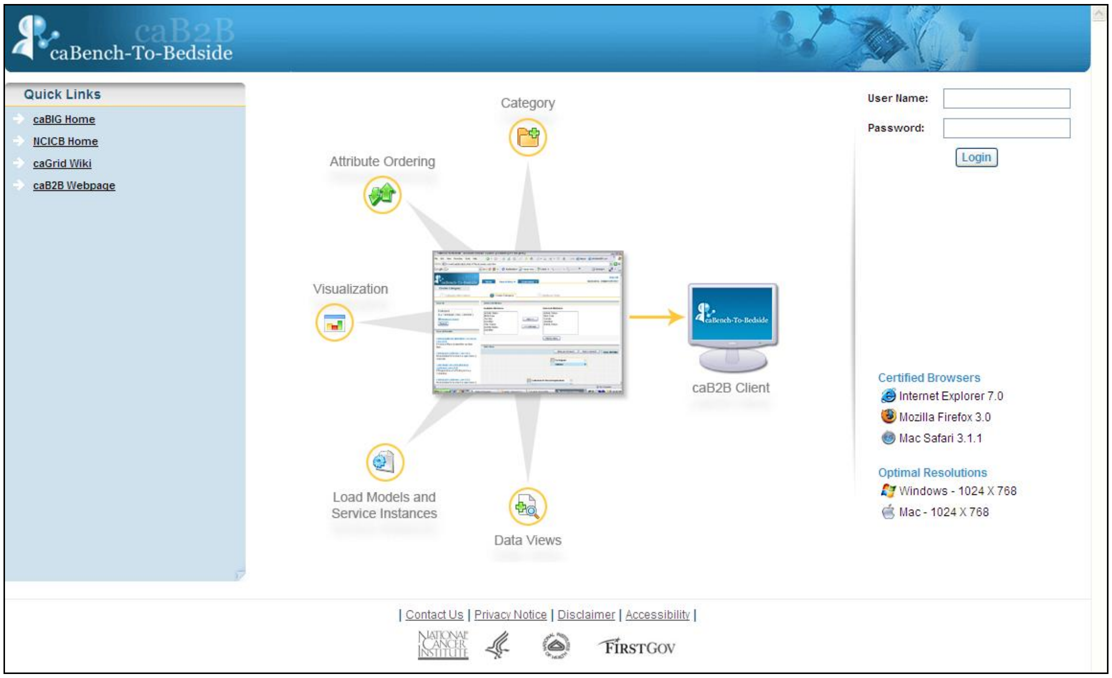


Figure 1.1. caB2B Administrative Module Login Page

You require a **User Name** and **Password** to log into the Administrative Module. You can get the login details from your caB2B installation owner. Once you log into the Administrative Module, you see the Administrative Module's home page shown below in Figure 1.2.

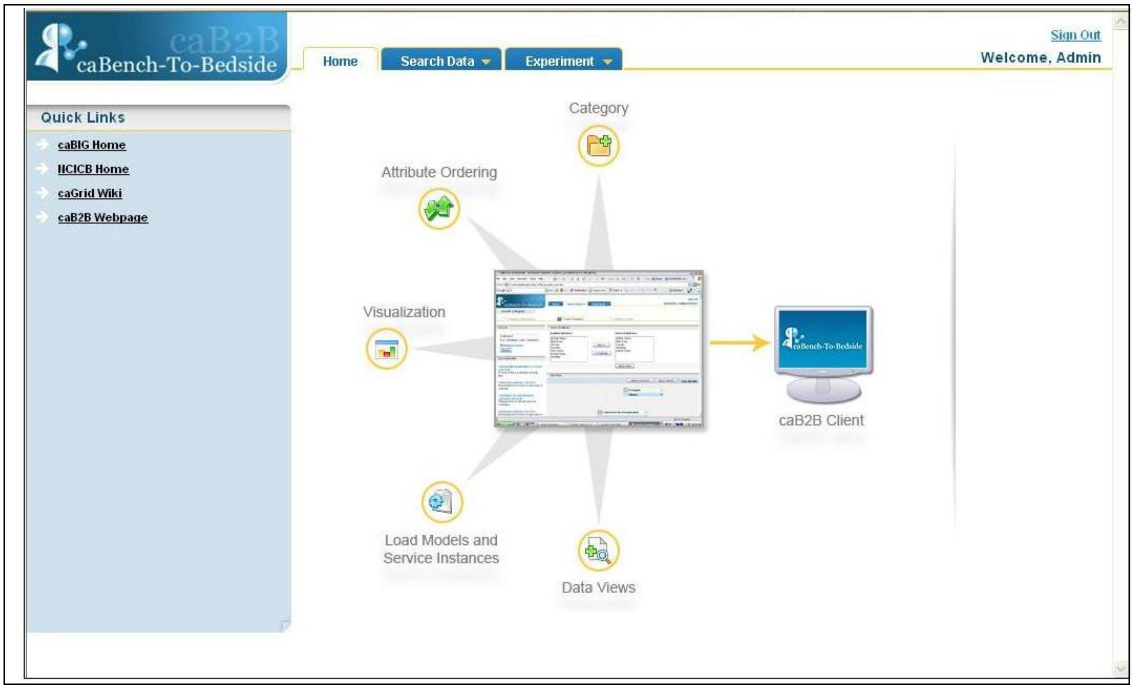


Figure 1.2. caB2B Administrative Module Home Page

Chapter 2 Load Models

This chapter describes details of how to load models. It contains the following topics:

- *Introduction to Load Models*
- *Steps to Load Model*
- *Using Loaded Models*

Introduction to Load Models

The Load Models functionality allows you to load models from the **ca**ncer **D**ata **S**tandards **R**epository (caDSR) into caB2B **M**eta**d**ata **R**epository (MDR). The caDSR is the central caBIG® repository used to store various models.

This functionality fetches all the models available in the caDSR and lets you load selected model into the caB2B MDR. Once you load a model into the caB2B MDR, the Client and the Web Application users can query the model. For example, loading the caArray model will enable end user to query for caArray classes like *Sample, Experiment* etc with attributes like *Experiment Title* and *Sample Name*.

What is a Model?

A model refers to a domain model (also known as a UML Model). A domain model is a conceptual model of a system that describes the various entities (along with their properties) involved in that system and the relationships between them. Thus, a model is a blueprint of the data stored in the system. For example, the domain model for caTissue service will have metadata regarding categories like participant with attributes first name, last name, gender etc.

What is Metadata Repository?

The Metadata Repository (MDR) stores the metadata used by caB2B. It consists of many models. Metadata of each model includes its domain objects, their properties, semantic annotations like Common Data Elements (CDEs) and permissible values. The MDR also contains all possible paths between each pair of entities. The caB2B server pre-calculates the paths between all pairs of entities in the model and stores them in the MDR.

Steps to Load Models

1. On the **Search Data** menu, click **Load Models**.

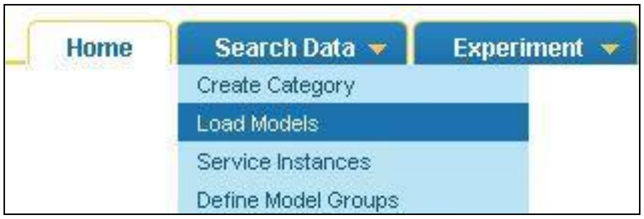
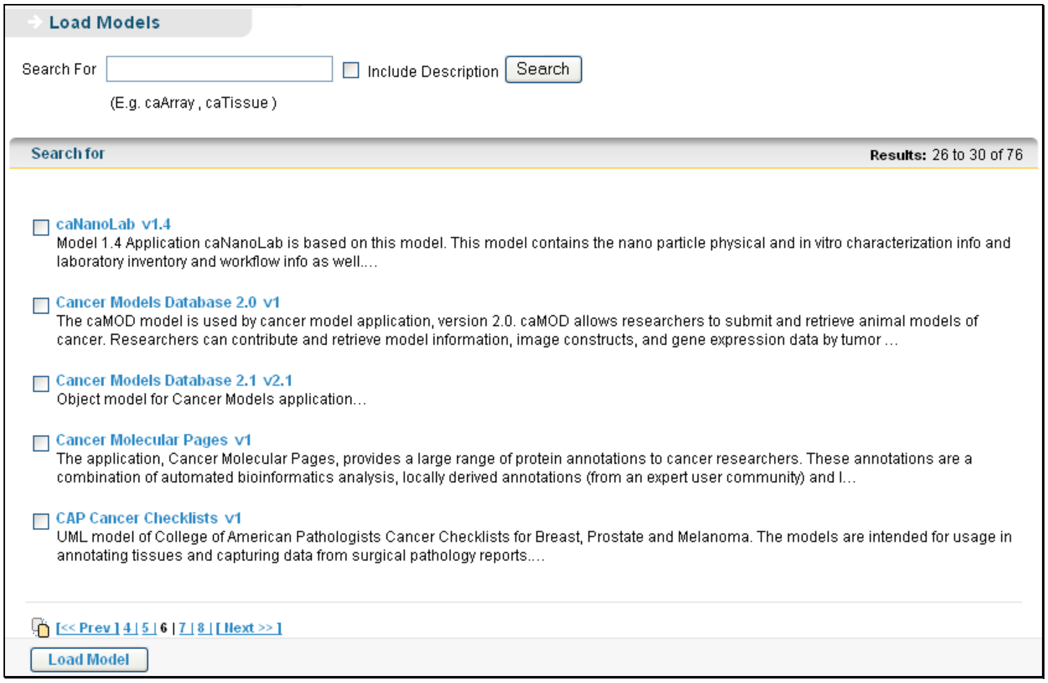


Figure 2.1. Search Data menu

2. The Load Model page opens, showing all available models in the caDSR, which have not been loaded in the caB2B MDR yet.



Load Models

Search For □ Include Description Search

(E.g. caArray , caTissue)

Search for — Results: 26 to 30 of 76

□ caNanoLab v1.4
Model 1.4 Application caNanoLab is based on this model. This model contains the nano particle physical and in vitro characterization info and laboratory inventory and workflow info as well....

□ Cancer Models Database 2.0 v1
The caMOD model is used by cancer model application, version 2.0. caMOD allows researchers to submit and retrieve animal models of cancer. Researchers can contribute and retrieve model information, image constructs, and gene expression data by tumor ...

□ Cancer Models Database 2.1 v2.1
Object model for Cancer Models application...

□ Cancer Molecular Pages v1
The application, Cancer Molecular Pages, provides a large range of protein annotations to cancer researchers. These annotations are a combination of automated bioinformatics analysis, locally derived annotations (from an expert user community) and l...

□ CAP Cancer Checklists v1
UML model of College of American Pathologists Cancer Checklists for Breast, Prostate and Melanoma. The models are intended for usage in annotating tissues and capturing data from surgical pathology reports....

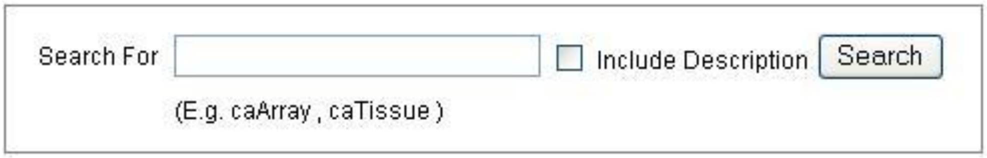
[<< Prev] 4 | 5 | 6 | 7 | 8 | [Next >>]

Load Model

Figure 2.2. Load Models Page

F.Y.I.	Models that are already present in MDR are not visible on Load Models page

3. To search for a model of your interest, enter a search string in the **Search For** box and click **Search**. If the search string is present in the model description, select the **Include Description** check box

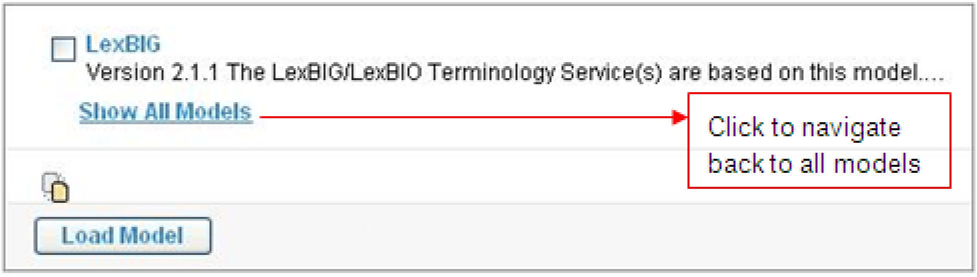


Search For □ Include Description Search

(E.g. caArray , caTissue)

Figure 2.3. Search for Models

4. Select the desired models from the visible models and click **Load Model** to start loading it. You can select more than one model to load them all.



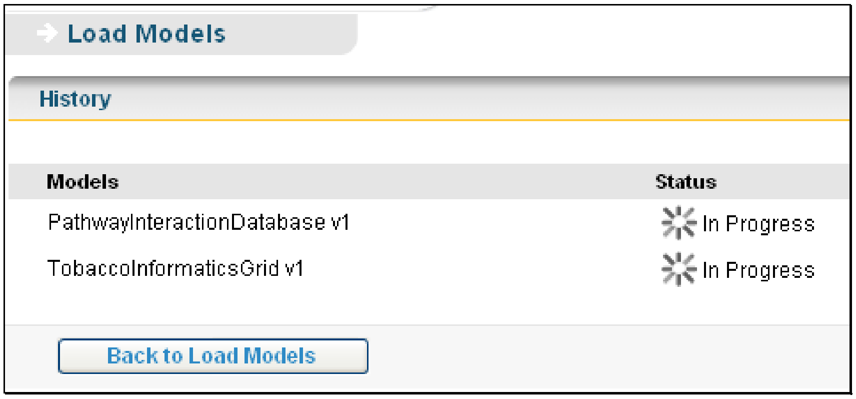
□ LexBIG
Version 2.1.1 The LexBIG/LexBIO Terminology Service(s) are based on this model....
Show All Models

Click to navigate back to all models

Figure 2.4. Click Load Model to load a model

F.Y.I.	To navigate back to view all matching models in the search result click **Show All Models**.

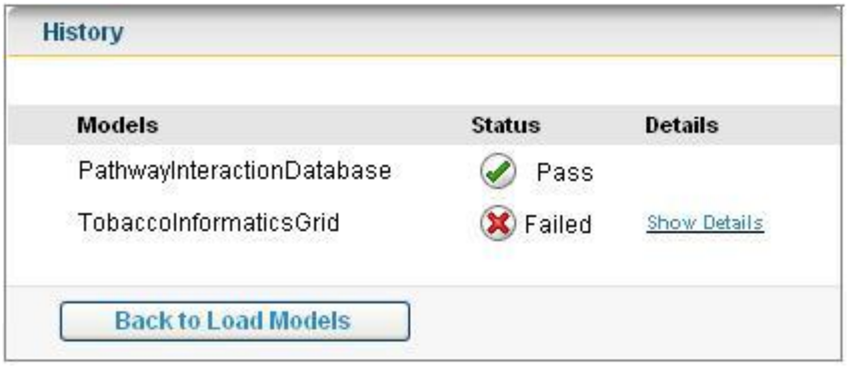
5. You will see a **History** page that shows progress of models you selected for loading.



Models	Status
PathwayInteractionDatabase v1	In Progress
TobaccoInformaticsGrid v1	In Progress

Figure 2.5. Model loading in progress

6. You can view the status once the loading is complete



Models	Status	Details
PathwayInteractionDatabase	✓ Pass	
TobaccoInformaticsGrid	✗ Failed	Show Details

Figure 2.6. Load Model History

F.Y.I.	You cannot remove a loaded model

In case, a particular model fails to load, you can view the reason by clicking **Show Details** link. Model loading generally fails due to the following reasons:

1. The model retrieved from caDSR is structurally incorrect, i.e. parser is unable to parse the model
2. The model retrieved from caDSR is empty, i.e. it does not have any entities.

Using Loaded Models

Once you load models, the users of Client Application and Web Application can use it. The Client Application user can view the models, by clicking the **My Settings** link on the home page. (Refer to *caB2B Client Application End User manual*)**.**

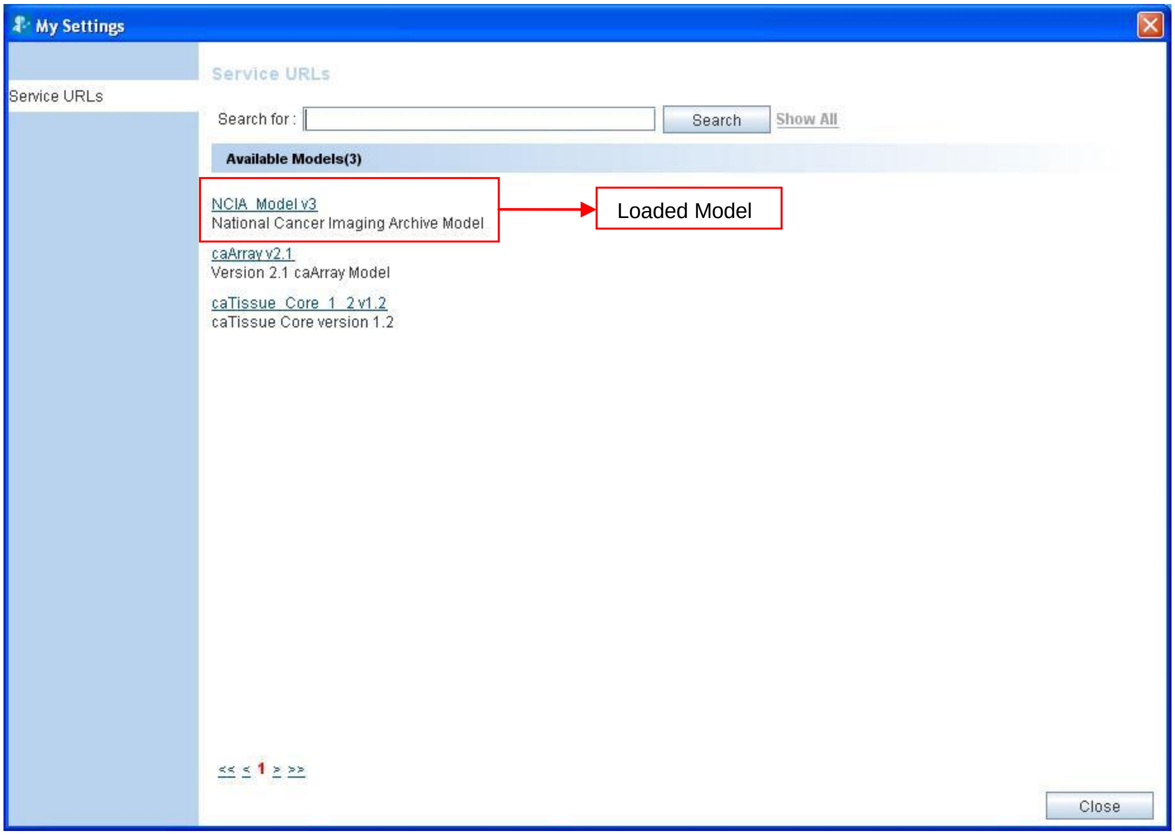


Figure 2.7. My Settings window showing the available models

The Web Application will not see the models but they will see the model groups (refer to *Define Model Groups*) created for these models. These model groups are visible in the **What type of data do you wish to search for?** dropdown in the Web Application.

Chapter 3 Define Model Groups

This chapter describes how to define model groups using different models. It contains the following topics:

- *Introduction to Model Groups*
- *Steps to define Model groups*
- *Using Model group*

Introduction to Model Groups

A model group represents one or more models that hold similar kind of data. They are meant to hide complexity of models like name, versions etc from end users. You can define model groups using any of the models present in caB2B MDR. The Web Application users can see these model groups on home page (Refer to *caB2B Web Application Manual).*

For example, you can define a model group *Biospecimen data* to represent models of tissue banking applications like *caTissuecore v1.2*, *caTissueSuite v1.1*. *Biospecimen data* clearly defines that it represents biospecimen data whereas this is not clear with the name *caTissue*. Therefore, model group creation would help the end user to query the desired data irrespective of the knowledge of model or the data service.

A model group created for a secure data service should also be secure. Secure data service is a service that allows only caGrid-authenticated users to access their data.

Steps to define Model groups

1. In the **Search data** menu, click **Define Model Groups.**

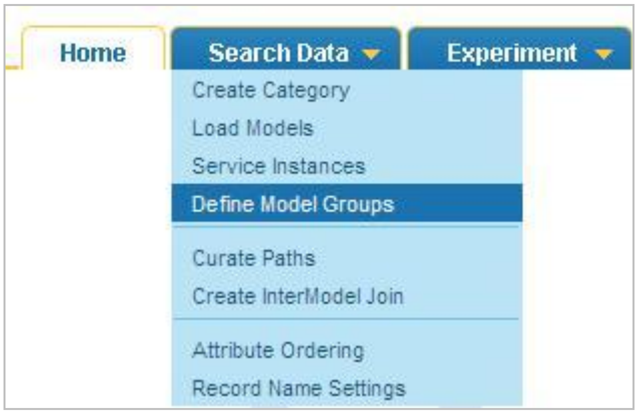
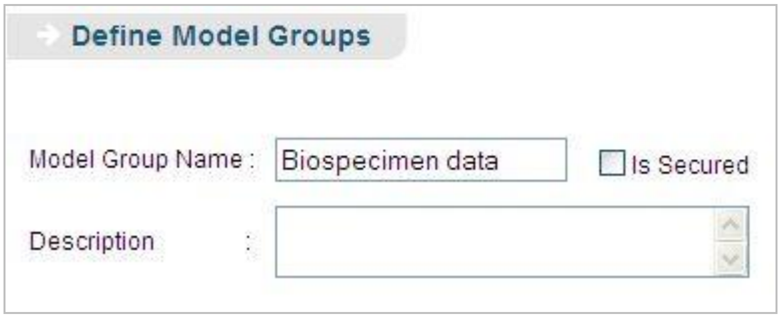


Figure 3.1. Define model group

2. Enter the **Model Group Name** and **Description.** Select **Is Secured** checkbox, if you are creating a model group for a secure data service, see *Figure 3.2*.



Define Model Groups

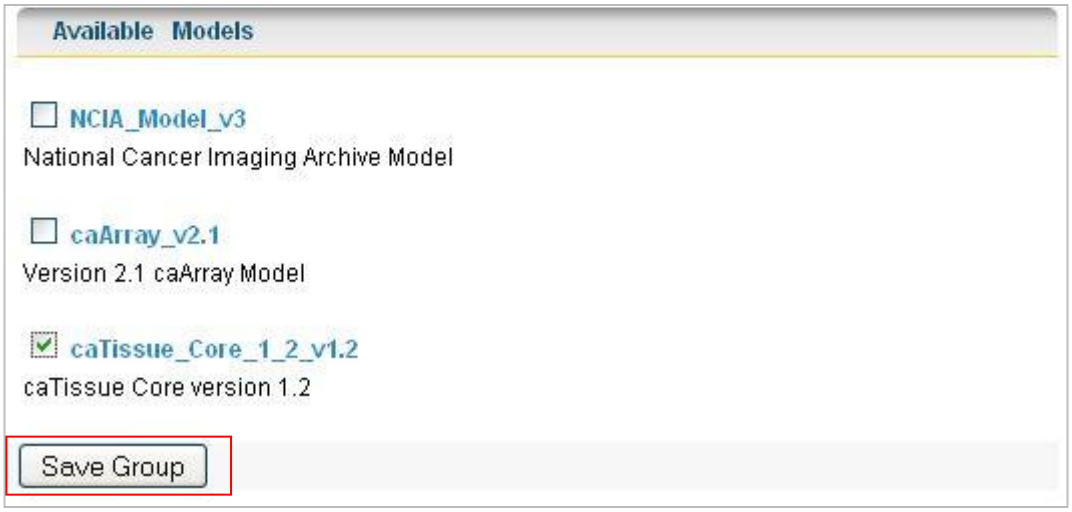
Model Group Name : Biospecimen data

□ Is Secured

Description :

Figure 3.2. Enter the name of model group

3. Select one or more models for which you wish to create the model group.



Available Models

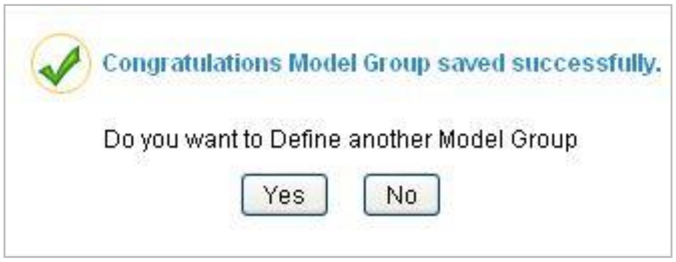
□ NCIA_Model_v3
National Cancer Imaging Archive Model

□ caArray_v2.1
Version 2.1 caArray Model

☑ caTissue_Core_1_2_v1.2
caTissue Core version 1.2

Figure 3.3. Select model from list

4. Click **Save Group**. You will see a message like below. Click **Yes** if you want to save another model group. Click **No** if you do not want to save another model group.



Congratulations Model Group saved successfully.

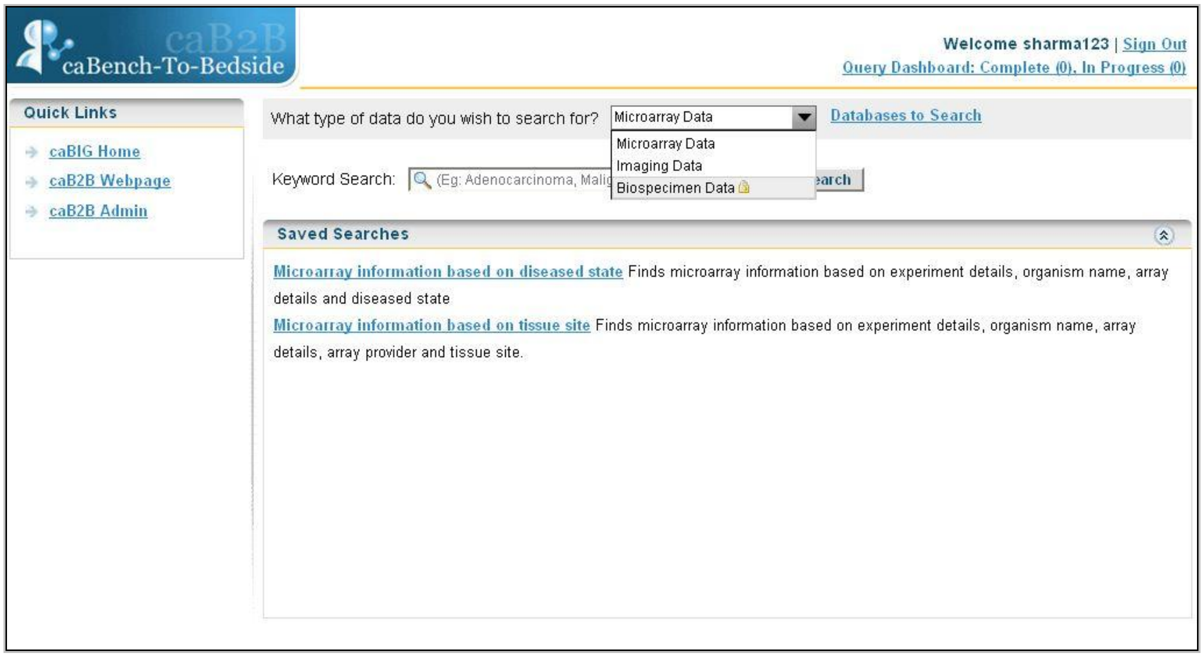
Do you want to Define another Model Group

Figure 3.4. Model group saved successfully

Using Model group

Once you create a model group, Web Application user will see that in the dropdown **What type of data you wish to search for?**

As explained in the introduction, the Web Application user can just select the type of data (Biospecimen data, Microarray data etc.) and execute a query without worrying about which application is queried, what is its metadata etc. In addition, user has to sign in before using secure model group.



What type of data do you wish to search for? Microarray Data

Keyword Search: (Eg: Adenocarcinoma, Malig

Microarray information based on diseased state Finds microarray information based on experiment details, organism name, array details and diseased state

Microarray information based on tissue site Finds microarray information based on experiment details, organism name, array details, array provider and tissue site.

Figure 3.5. Select Model group in caB2B Web Application

Chapter 4 Service Instances

This chapter describes how to configure the service instances. It contains the following topics:

- *Introduction to Service Instances*
- *Steps to Configure* Service Instances
- *Using Service Instances*

Introduction to Service Instances

Service instances are caGrid-enabled services that returns data based on the query. Every service instance has a model associated with it that is same as the corresponding domain model stored in caDSR.

When a model is present in caB2B MDR, you can use service instances functionality to find all active services that understands the particular model. Then you can define default services that will be queried when user does not select any while defining the query.

F.Y.I.	User can override the service instances you have selected by instances s/he wish to query

You can select multiple service instances to query for a model. For example, you can configure the Washington University's and the University of Pittsburgh's caTissue services to query. We recommend that you should define service instances for each model present in caB2B MDR.

Steps to Configure Service Instances

To configure the service instance, you must load at least one model. After you have loaded model(s), follow the steps given below:

1. On the **Search Data** menu, click **Service Instances**.

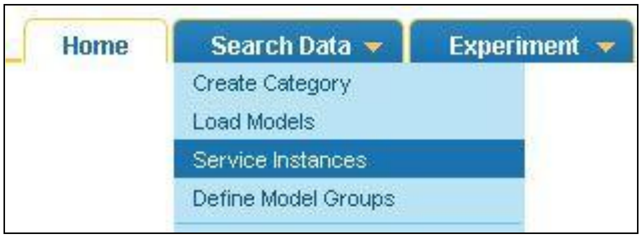
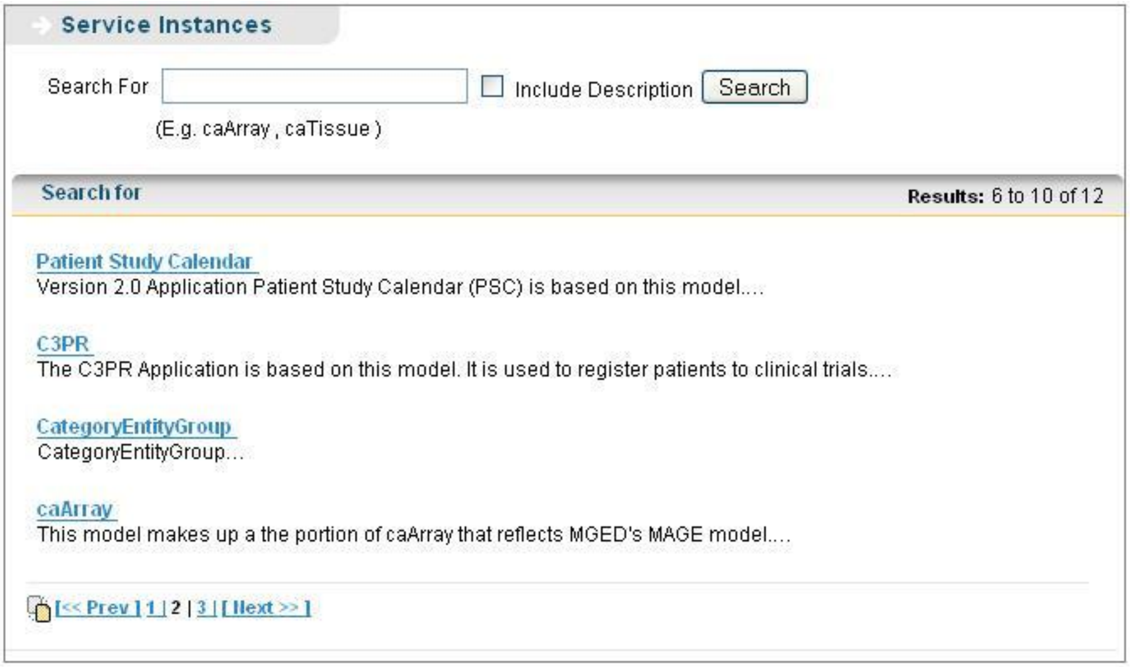


Figure 4.1. Search Data menu

2. The Service Instances page shows all loaded models.



Service Instances

Search For □ Include Description Search

(E.g. caArray , caTissue)

Search for — Results: 6 to 10 of 12

Patient Study Calendar
Version 2.0 Application Patient Study Calendar (PSC) is based on this model....

C3PR
The C3PR Application is based on this model. It is used to register patients to clinical trials....

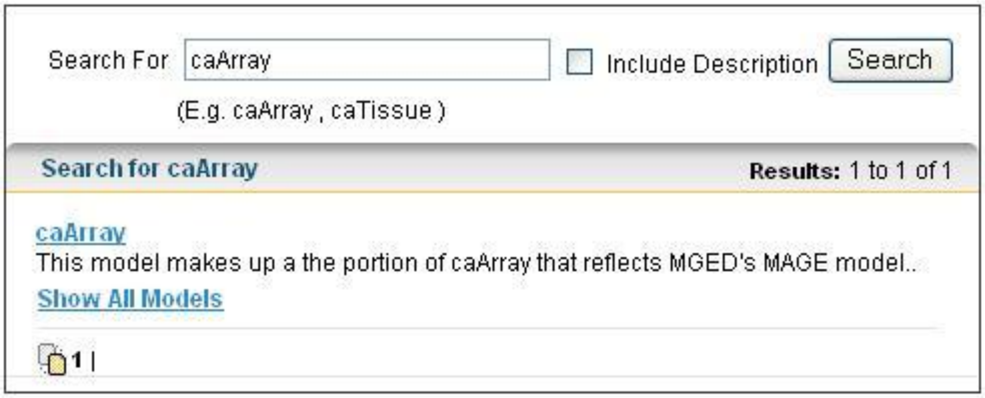
CategoryEntityGroup
CategoryEntityGroup...

caArray
This model makes up a the portion of caArray that reflects MGED's MAGE model....

[<< Prev] 1 | 2 | 3 | [Next >>]

Figure 4.2. Service Instances Page

3. In the **Search For** box, enter a search string and click **Search**. Select the **Include Description** check box, if the search string is present in the model description.



Search For: caArray □ Include Description Search

(E.g. caArray , caTissue)

Search for caArray — Results: 1 to 1 of 1

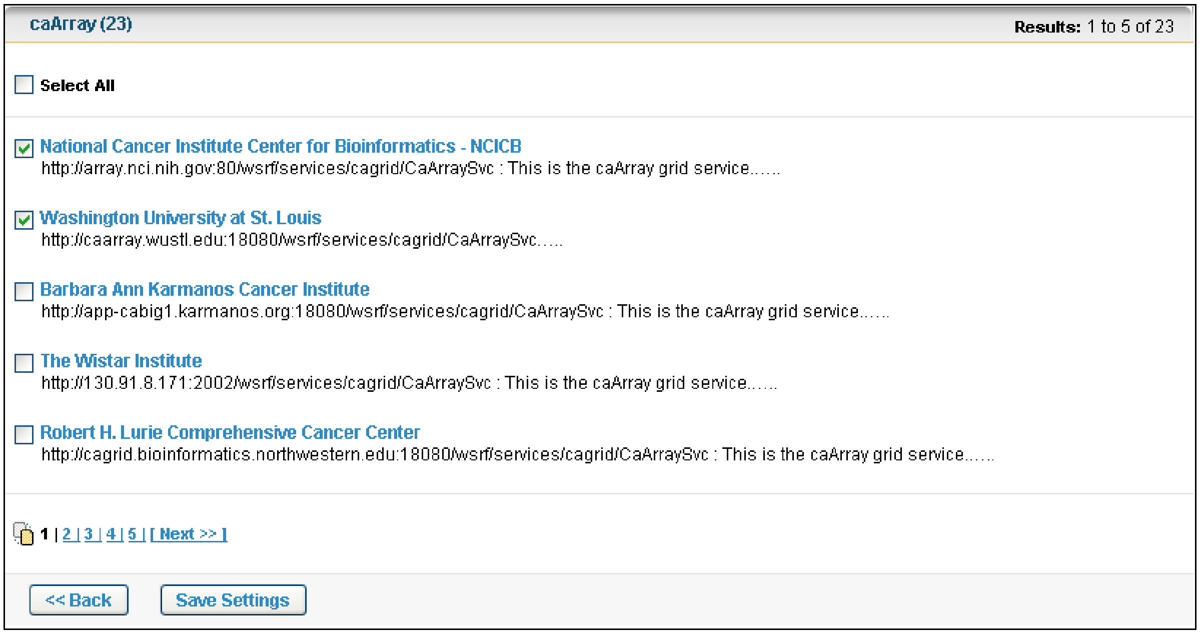
caArray
This model makes up a the portion of caArray that reflects MGED's MAGE model..

Show All Models

1 |

Figure 4.3. Searching a model

4. Click the model name to view its available services.
5. Select/Deselect the available services.



caArray (23) **Results:** 1 to 5 of 23

□ **Select All**

☑ National Cancer Institute Center for Bioinformatics - NCICB
http://array.nci.nih.gov:80/wsrf/services/cagrid/CaArraySvc : This is the caArray grid service......

☑ Washington University at St. Louis
http://caarray.wustl.edu:18080/wsrf/services/cagrid/CaArraySvc.....

□ Barbara Ann Karmanos Cancer Institute
http://app-cabig1.karmanos.org:18080/wsrf/services/cagrid/CaArraySvc : This is the caArray grid service......

□ The Wistar Institute
http://130.91.8.171:2002/wsrf/services/cagrid/CaArraySvc : This is the caArray grid service......

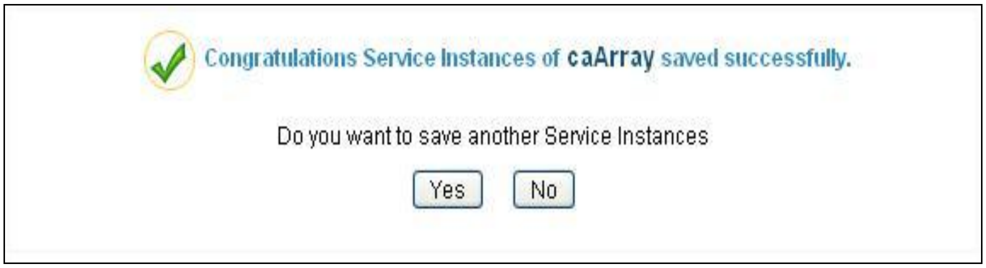
□ Robert H. Lurie Comprehensive Cancer Center
http://cagrid.bioinformatics.northwestern.edu:18080/wsrf/services/cagrid/CaArraySvc : This is the caArray grid service......

Figure 4.4. Service Instances currently available for the model

F.Y.I.

Click Back to navigate back to the Service Instances page without saving any of the instances

6. Click **Save Settings**. You see a success message if the service is saved. Click **Yes** in the message window to save another service instance, or click **No** to exit.

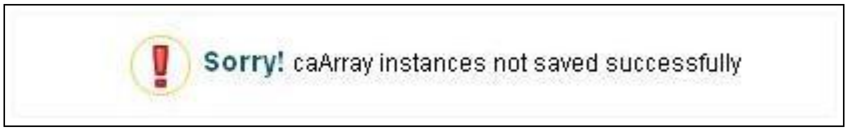


Congratulations Service Instances of **caArray** saved successfully.

Do you want to save another Service Instances

Figure 4.5. Success Message

7. You see a failure message if the service is not saved.



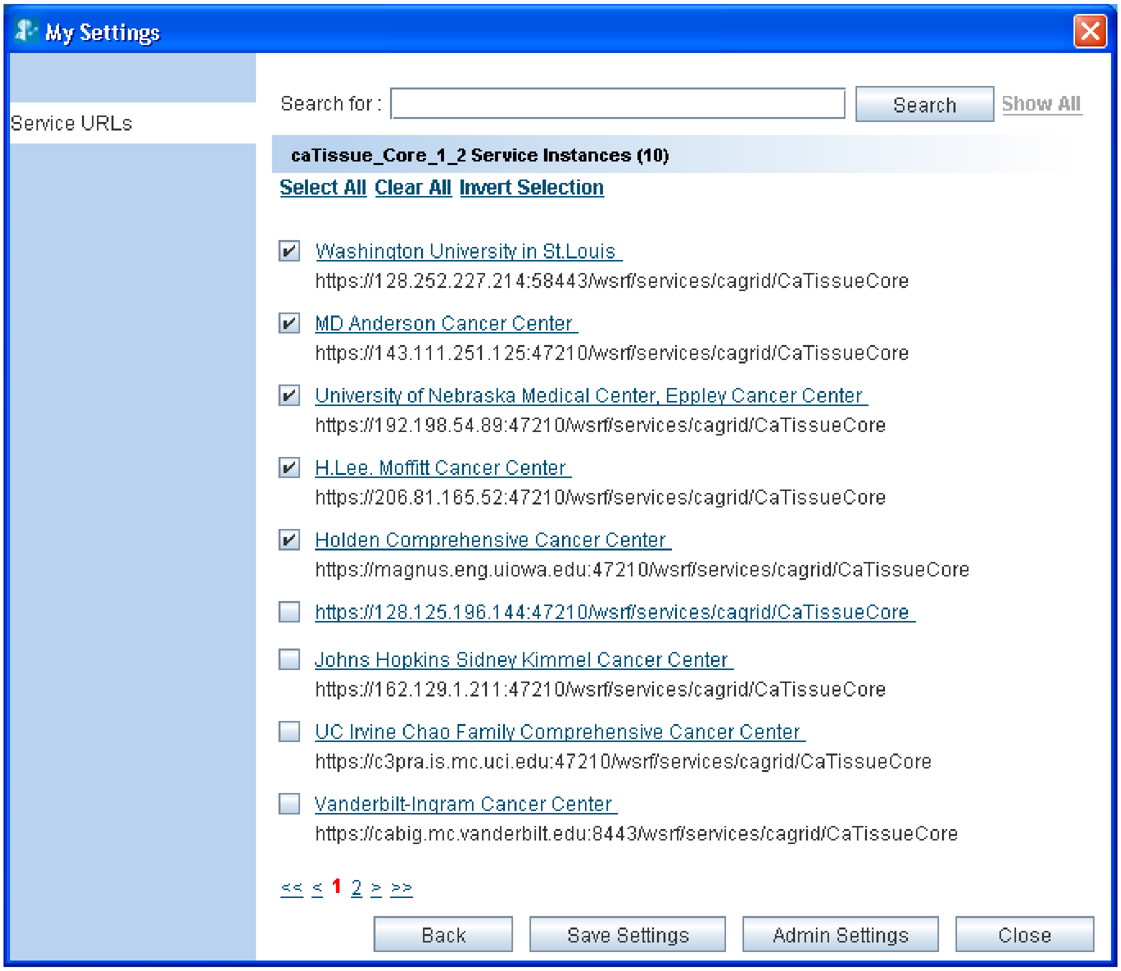
Sorry! caArray instances not saved successfully

Figure 4.6. Failure message

Using Service Instances

You can configure the service instances for the Client Application and Web Application. By default, a user would see the administrator-defined settings. User can change the settings from the **My Settings** link in Client Application home page or during the third step of query (refer to *caB2B Client Application End User manual*)

A Client Application user can see the administrator-defined services as selected.



caTissue_Core_1_2 Service Instances (10)

☑ Washington University in St.Louis
https://128.252.227.214:58443/wsrf/services/cagrid/CaTissueCore

☑ MD Anderson Cancer Center
https://143.111.251.125:47210/wsrf/services/cagrid/CaTissueCore

☑ University of Nebraska Medical Center, Eppley Cancer Center
https://192.198.54.89:47210/wsrf/services/cagrid/CaTissueCore

☑ H.Lee. Moffitt Cancer Center
https://206.81.165.52:47210/wsrf/services/cagrid/CaTissueCore

☑ Holden Comprehensive Cancer Center
https://magnus.eng.uiowa.edu:47210/wsrf/services/cagrid/CaTissueCore

□ https://128.125.196.144:47210/wsrf/services/cagrid/CaTissueCore

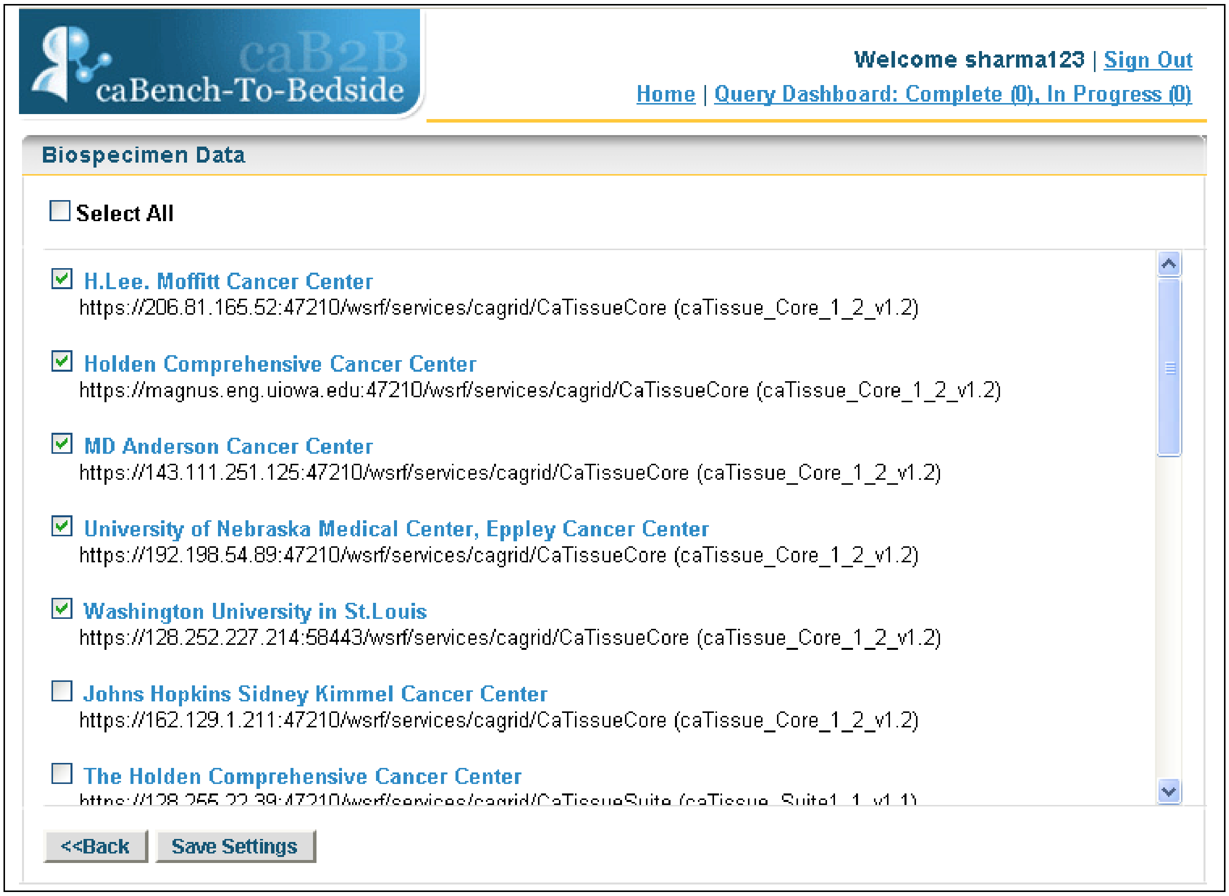
□ Johns Hopkins Sidney Kimmel Cancer Center
https://162.129.1.211:47210/wsrf/services/cagrid/CaTissueCore

□ UC Irvine Chao Family Comprehensive Cancer Center
https://c3pra.is.mc.uci.edu:47210/wsrf/services/cagrid/CaTissueCore

□ Vanderbilt-Ingram Cancer Center
https://cabig.mc.vanderbilt.edu:8443/wsrf/services/cagrid/CaTissueCore

Figure 4.7. Administrator-defined service instances

In Web Application, the services are available in **Databases to Search.** Users can view the administrator-defined service instances and configure the settings by clicking **Databases to Search** on home page.



Welcome sharma123 | Sign Out

Home | Query Dashboard: Complete (0), In Progress (0)

Biospecimen Data

□ Select All

☑ H.Lee. Moffitt Cancer Center
https://206.81.165.52:47210/wsrf/services/cagrid/CaTissueCore (caTissue_Core_1_2_v1.2)

☑ Holden Comprehensive Cancer Center
https://magnus.eng.uiowa.edu:47210/wsrf/services/cagrid/CaTissueCore (caTissue_Core_1_2_v1.2)

☑ MD Anderson Cancer Center
https://143.111.251.125:47210/wsrf/services/cagrid/CaTissueCore (caTissue_Core_1_2_v1.2)

☑ University of Nebraska Medical Center, Eppley Cancer Center
https://192.198.54.89:47210/wsrf/services/cagrid/CaTissueCore (caTissue_Core_1_2_v1.2)

☑ Washington University in St.Louis
https://128.252.227.214:58443/wsrf/services/cagrid/CaTissueCore (caTissue_Core_1_2_v1.2)

□ Johns Hopkins Sidney Kimmel Cancer Center
https://162.129.1.211:47210/wsrf/services/cagrid/CaTissueCore (caTissue_Core_1_2_v1.2)

□ The Holden Comprehensive Cancer Center
https://128.255.22.39:47210/wsrf/services/cagrid/CaTissueSuite (caTissue_Suite1_1_v1.1)

Figure 4.8. Administrator-defined service instances in caB2B Web Application

Chapter 5 Create Category

This chapter describes how to create categories. It contains the following topics:

- *Introduction to Category*
- *Introduction to Path*
- *Steps to Create a Category*
- *Use of Category*

Introduction to Category

A category is a collection of *searchable* attributes from different classes. You can use a category to search, view, and add data to experiments from Client Application of caB2B. A category can be a combination of several entities or domain objects or classes. For example:

1. Participant, a domain object in the caTissue Core application, is a class consisting of the following attributes:
    - Identifier
    - First Name
    - Last Name
    - Gender
    - Race
2. Gene Annotation, a domain object in the GeneConnect application, is a class consisting of the following attributes:
    - Entrez Gene ID
    - Gene Symbol
    - Gene Name
    - GenBank Accession Number
    - Taxonomy ID
    - Chromosome

The classes Participant and Gene Annotation in the examples above refer to a single class respectively. You can use Create Category feature to define a category that combine two or more classes, like participant and specimen. You can customize a category to have selected attributes that are useful in queries instead of all attributes from all constituent classes.

For example, an end-user wants to get all participants whose specimens are available. Since specimen availability is not an attribute of the Participant class, you can create a category, in which the specimen-availability attribute from the Specimen class and the attributes from the Participant class.

Introduction to Path

A path represents list of relationships that connect two classes present in a model. There can be multiple paths connecting two classes. The results of a query differ based on the path you select. Therefore, it is important to choose the right path while creating the category. For example, consider the two entities *Specimen* and *Site* (from the caTissue Core application). A *Specimen* is associated with the *Site* class in two ways:

1. Site where the specimen was collected
2. Site where the specimen is stored

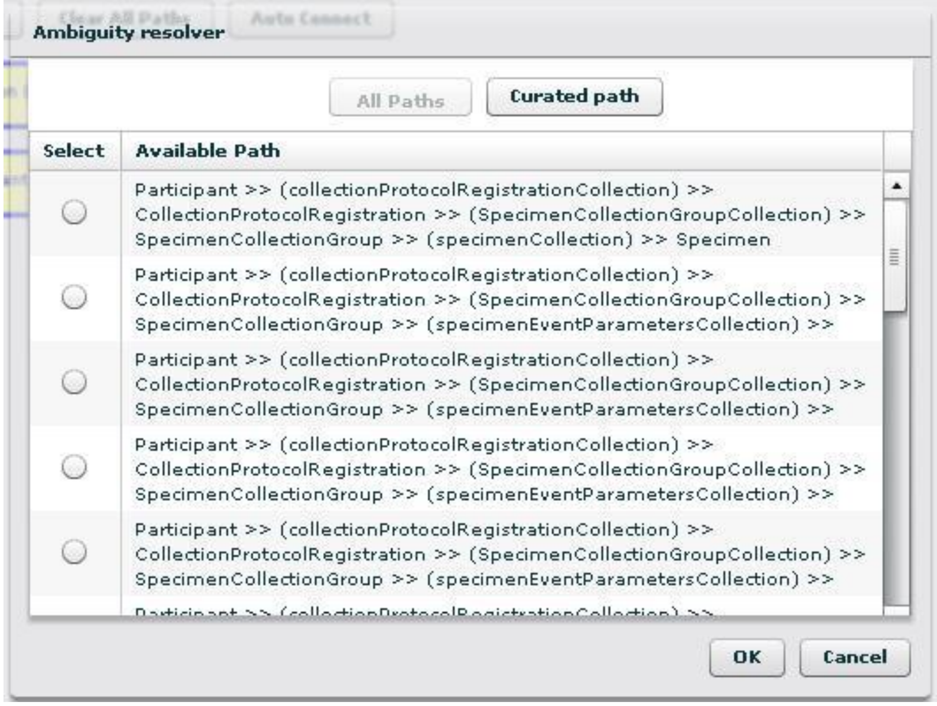
That is, there are two paths between the class *Specimen* and *Site*: 1) Through *Specimen Collection Group* for the collection site, 2) Through *Storage Container*

Query example 1: Show all the available specimens collected in Barnes Jewish Hospital. Since you are specifically interested in a particular collection site in this query, you must choose the path through Specimen Collection Group.

Query example 2: Show all the available specimens stored in Barnes Jewish Hospital. Since you are specifically interested in a particular storage site in this query, you must choose the path through Storage Container.

What is an Ambiguity Resolver?

There are certain classes between which multiple paths exist. Depending on the category, you may wish to choose different paths. In such a situation Ambiguity Resolver window appears when you try to connect the two categories. You can select one of the available paths from the Ambiguity Resolver window to connect the two classes.



Select	Available Path
○	Participant >> (collectionProtocolRegistrationCollection) >> CollectionProtocolRegistration >> (SpecimenCollectionGroupCollection) >> SpecimenCollectionGroup >> (specimenCollection) >> Specimen
○	Participant >> (collectionProtocolRegistrationCollection) >> CollectionProtocolRegistration >> (SpecimenCollectionGroupCollection) >> SpecimenCollectionGroup >> (specimenEventParametersCollection) >>
○	Participant >> (collectionProtocolRegistrationCollection) >> CollectionProtocolRegistration >> (SpecimenCollectionGroupCollection) >> SpecimenCollectionGroup >> (specimenEventParametersCollection) >>
○	Participant >> (collectionProtocolRegistrationCollection) >> CollectionProtocolRegistration >> (SpecimenCollectionGroupCollection) >> SpecimenCollectionGroup >> (specimenEventParametersCollection) >>
○	Participant >> (collectionProtocolRegistrationCollection) >> CollectionProtocolRegistration >> (SpecimenCollectionGroupCollection) >> SpecimenCollectionGroup >> (specimenEventParametersCollection) >>
	Participant >> (collectionProtocolRegistrationCollection) >>

Figure 5.1. Ambiguity Resolver window

Steps to Create a Category

1. On the **Search Data** menu, click **Create Category**.

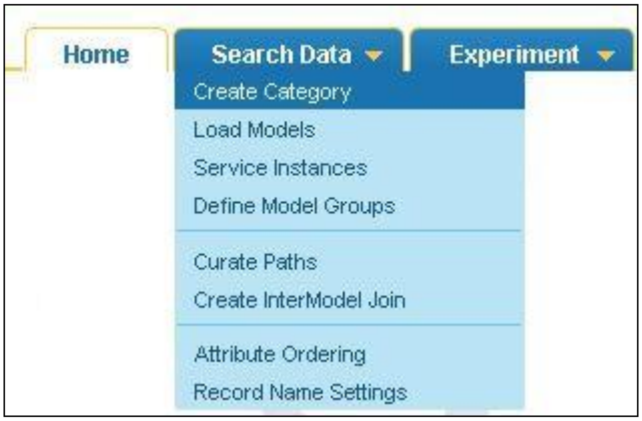


Figure 5.2. Search Data menu

2. The Create Category page appears, showing three tabs: **Category Information**, **Create Category**, and **Attribute Order**.

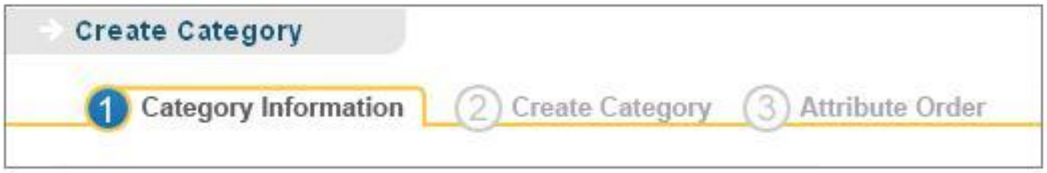


Figure 5.3. Category Creation process

3. In the **Category Information** tab, specify the **Category Title** and an optional Category **Description.** Then click **Next>>**

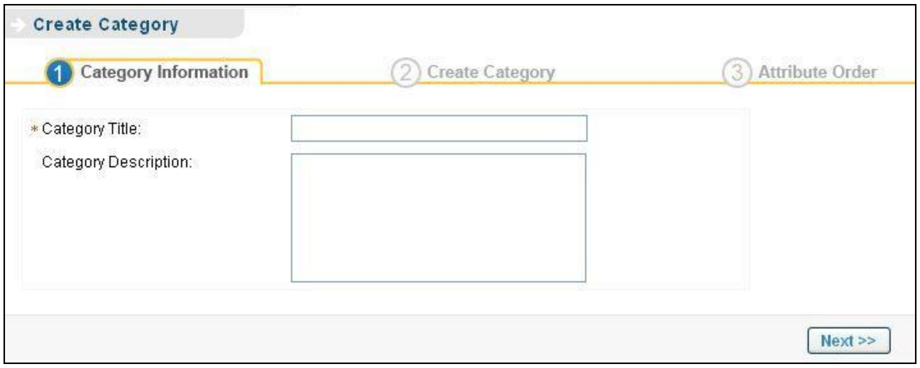
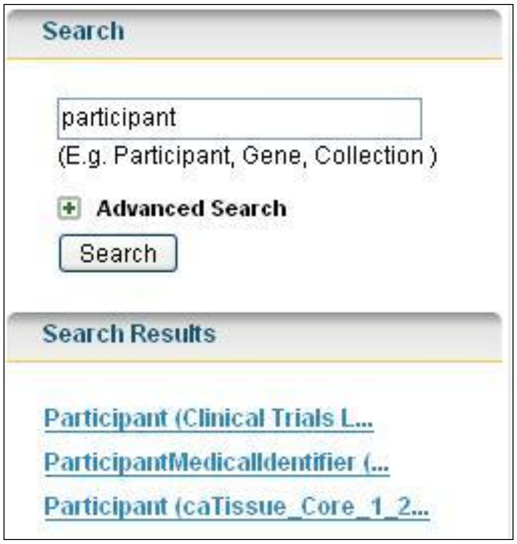


Figure 5.4. Category Information tab

4. In the **Create Category** tab, perform the following steps:

   a. Enter a search string and click **Search**.



participant

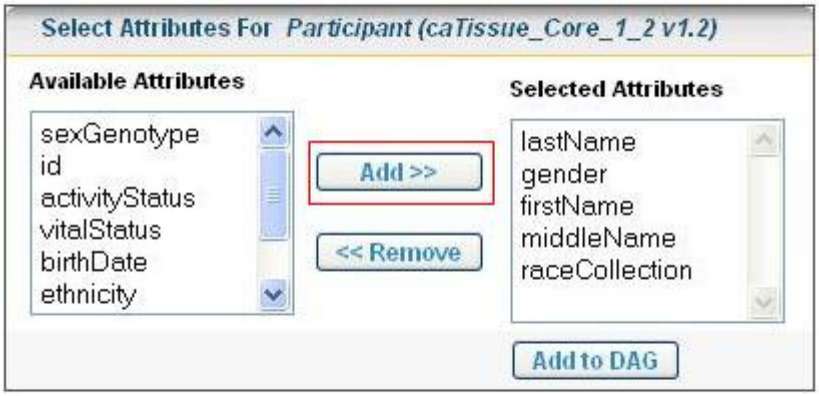
Participant (Clinical Trials L...
ParticipantMedicalIdentifier (...
Participant (caTissue_Core_1_2...

Figure 5.5. Searching for metadata

F.Y.I.	Choose the **Advanced Search** option to refine your search results. To know more about the advanced search options, refer to *Appendix A - Advanced Search.*

b. In the **Search Results**, click the class you want to use. The **Available Attributes** list shows the attributes from the class.

c. In the **Available Attributes** list, select the attributes you want in the category (To select multiple attributes, hold down the CTRL key while selecting the attributes). Click **Add** to add the attributes to **Selected Attributes** list.



sexGenotype
id
activityStatus
vitalStatus
birthDate
ethnicity

lastName
gender
firstName
middleName
raceCollection

Figure 5.6. The Entity Panel

F.Y.I.	In the **Selected Attributes** list, select the attributes to remove from the category. Click **Remove** to remove the attributes from the category.

d. Click **Add to DAG** to add the category to **D**i**AG**rammatic View.

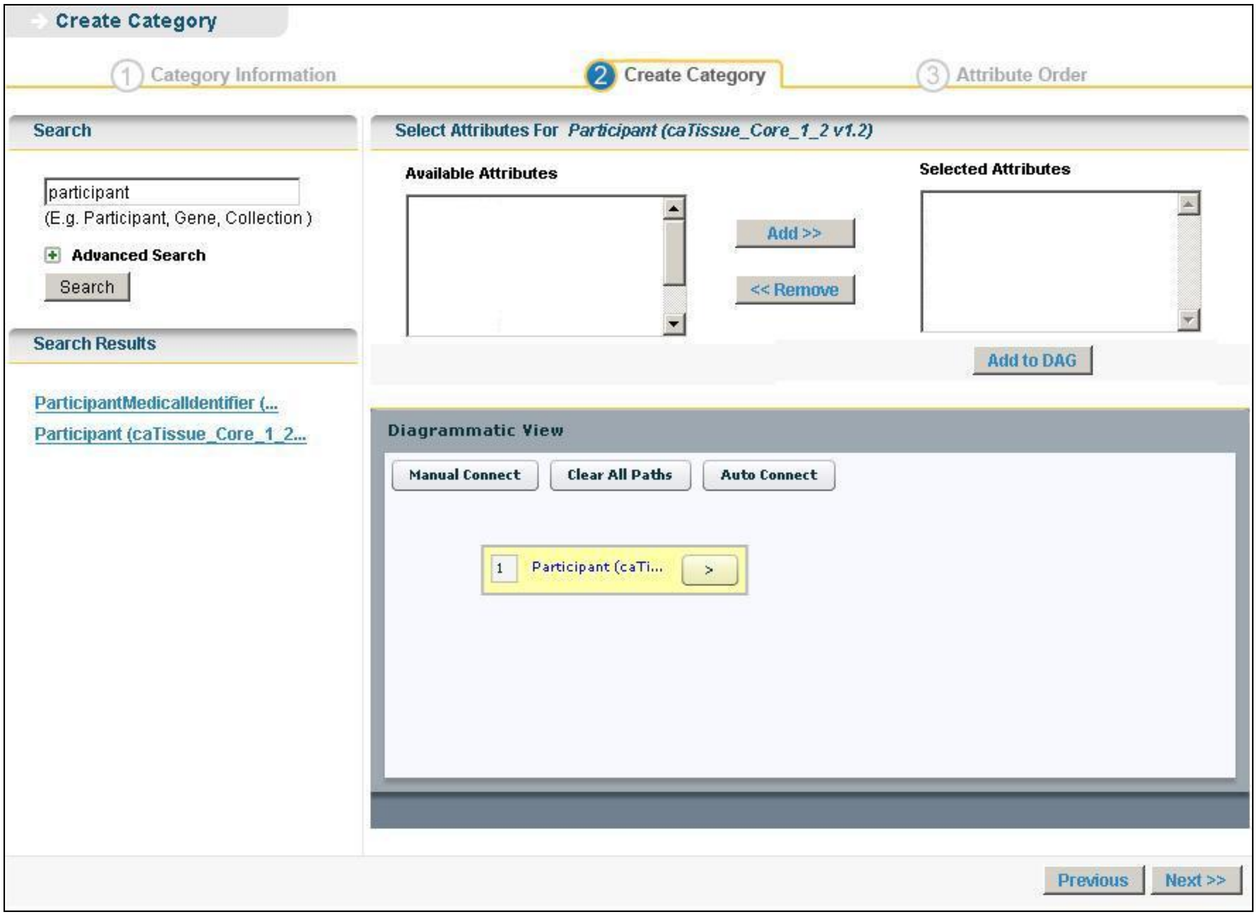
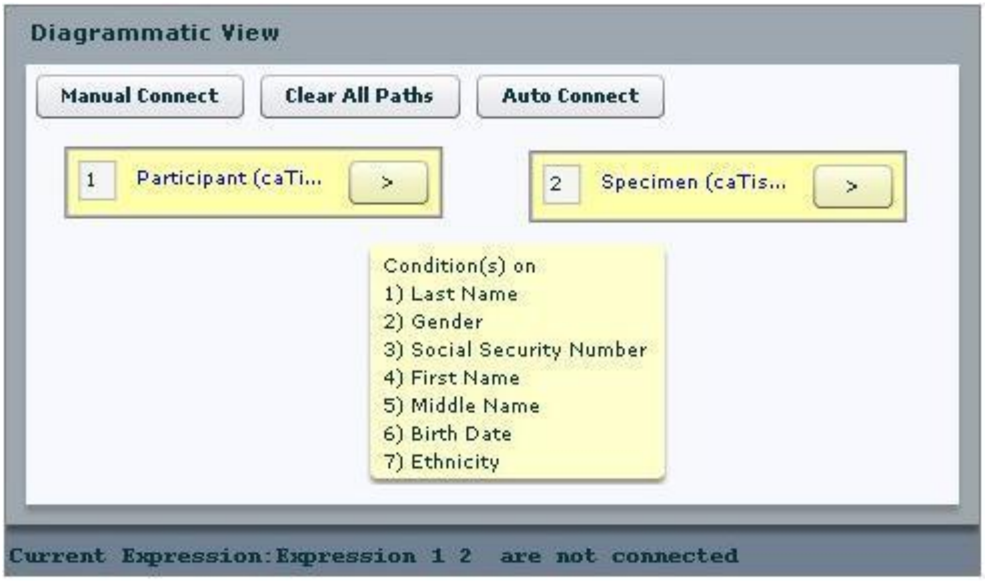


Figure 5.7. The DAG Panel

F.Y.I.	The **DiAG**rammatic View (DAG) panel is similar to a canvas where all the classes in a category appear in the form of nodes. Application adds a yellow node for each class that you add. As you continue to add classes, application adds corresponding node to the DAG view.

e. Repeat the steps (a) to (d) to add more classes. Clicking the arrow > gives options to Edit or Delete the class added to DAG. After adding all the classes, the DAG will look similar to the figure below.

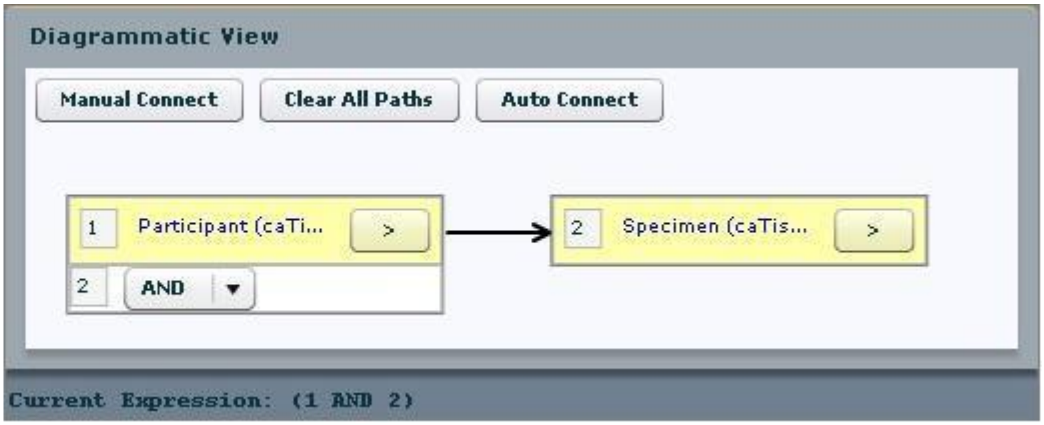


Condition(s) on
1) Last Name
2) Gender
3) Social Security Number
4) First Name
5) Middle Name
6) Birth Date
7) Ethnicity

Current Expression:Expression 1 2 are not connected

Figure 5.8. Diagrammatic View

f. Click **Manual Connect** to connect the classes. If multiple paths exist between the classes, then you have to select a path from the Ambiguity Resolver window (see *Figure 5.1*).

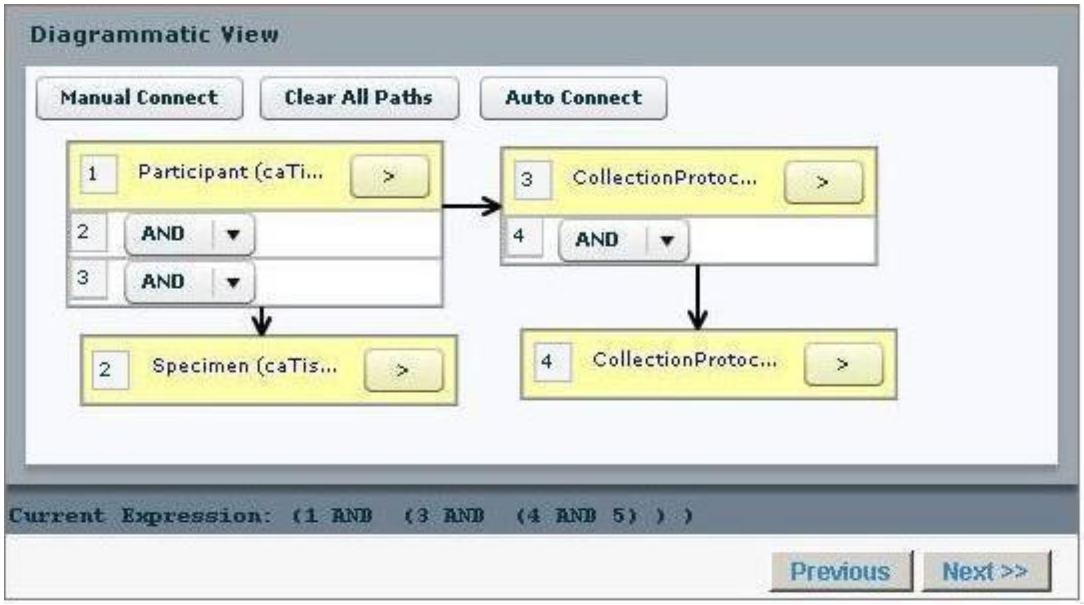


Current Expression: (1 AND 2)

Figure 5.9. DAG with nodes connected

F.Y.I.	Click **Auto Connect** to connect the classes when a curated path exists(see *Use of the Curated Path*)

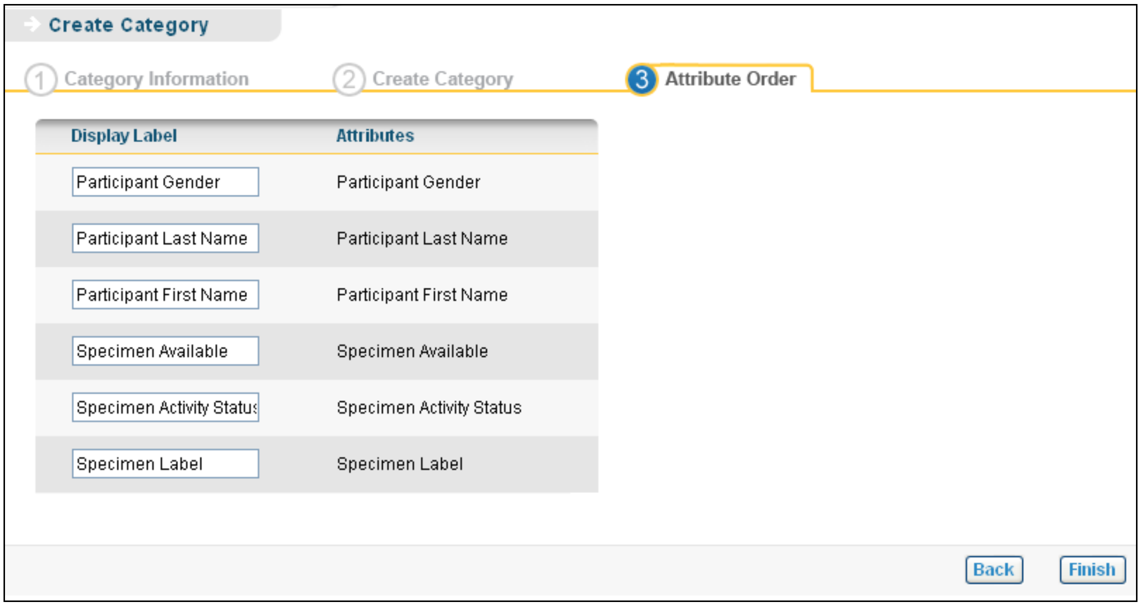
g. Once all classes are connected, click **Next** to navigate to the **Attribute Order** tab.



Current Expression: (1 AND (3 AND (4 AND 5)))

Figure 5.10. Four nodes connected in DAG

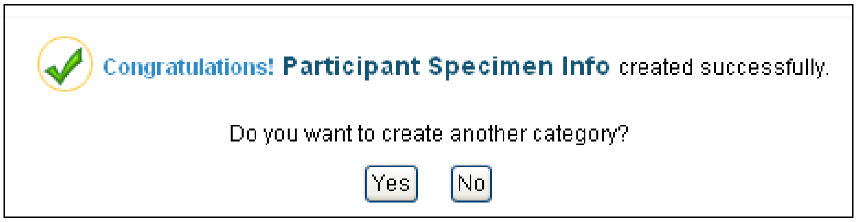
5. The **Attribute Order** tab displays all the attributes added to the category. You can change the display label. Click **Finish** to save the category.



Display Label	Attributes
Participant Gender	Participant Gender
Participant Last Name	Participant Last Name
Participant First Name	Participant First Name
Specimen Available	Specimen Available
Specimen Activity Statu:	Specimen Activity Status
Specimen Label	Specimen Label

Figure 5.11. Attribute Order tab

6. A success message appears on successful category creation. Click **Yes** in the message to create a new category, or click **No** to exit.

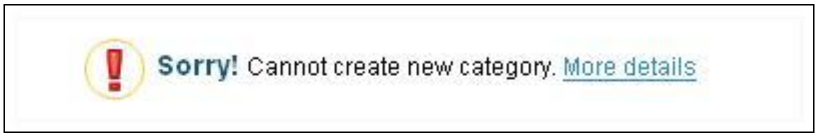


Congratulations! **Participant Specimen Info** created successfully.

Do you want to create another category?

Figure 5.12. Success Message

7. An error message appears if the category creation fails.



Sorry! Cannot create new category. More details

Figure 5.13. Failure Message

Category Creation Rules

As you go on connecting classes in DAG, you will find that you are building a graph (refer to http://en.wikipedia.org/wiki/Graph_(mathematics) for more details). The graph of your category should have single root i.e. the graph should be a tree (refer to http://en.wikipedia.org/wiki/Tree_(graph_theory) for more details)

For example, while creating the above category, there are two ways in which you can connect the classes.
Set-1:

- Participant>Collection protocol registration
- Collection protocol registration >Specimen collection group
- Specimen collection group>Specimen

Set-2:

- Participant>Collection protocol registration
- Specimen>Specimen collection group

Set-1 would create the category as it forms a tree with single root class (Participant).
Set-2 would give rise to a category with two root classes. Therefore, application will not save this category. You should always ensure that the category follows the rules specified above.

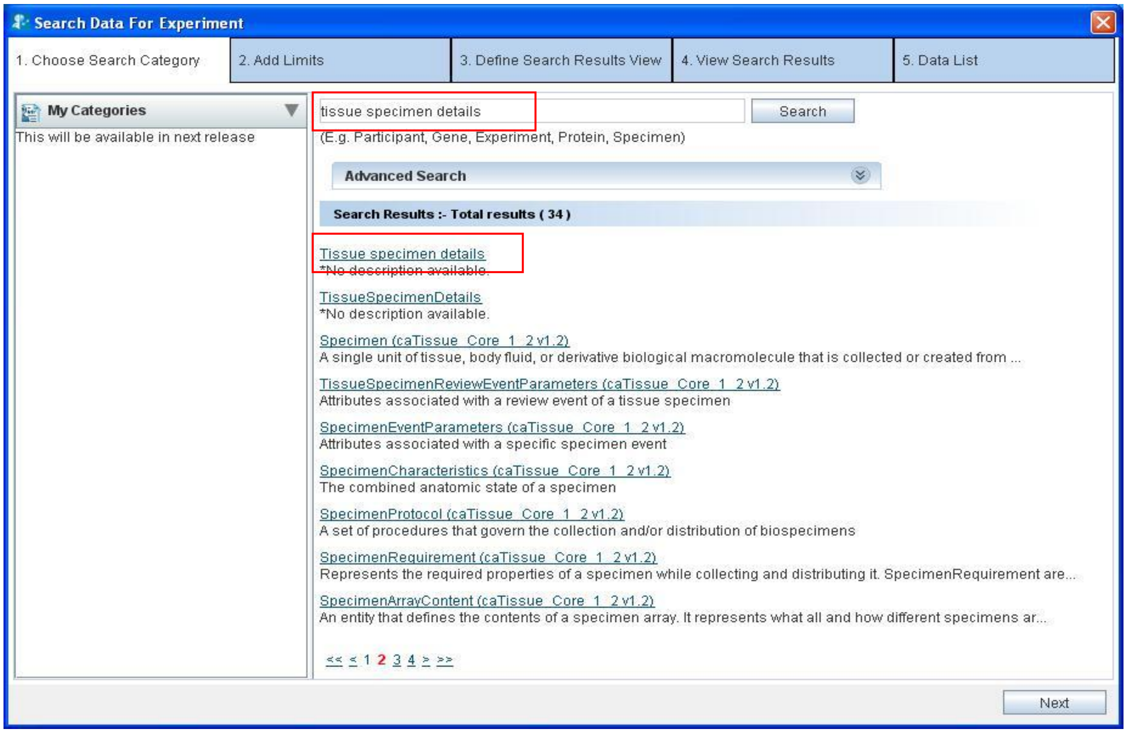
Use of Category

You can create categories to make the querying easier for the users of Client and the Web Application. For example, an end user wants to query *"All tissue specimens from female participants"*. You can create a category "*Tissue specimen details*" as explained above.

Steps to use category in Client Application:

1. Launch Client Application.
2. Enter the search string as "*Tissue specimen details".*

3. Click **Search**



Search Data For Experiment

1. Choose Search Category | 2. Add Limits | 3. Define Search Results View | 4. View Search Results | 5. Data List

My Categories

This will be available in next release

tissue specimen details

Search

(E.g. Participant, Gene, Experiment, Protein, Specimen)

Advanced Search

Search Results :- Total results (34)

Tissue specimen details
*No description available.

TissueSpecimenDetails
*No description available.

Specimen (caTissue_Core_1_2 v1.2)
A single unit of tissue, body fluid, or derivative biological macromolecule that is collected or created from ...

TissueSpecimenReviewEventParameters (caTissue_Core_1_2 v1.2)
Attributes associated with a review event of a tissue specimen

SpecimenEventParameters (caTissue_Core_1_2 v1.2)
Attributes associated with a specific specimen event

SpecimenCharacteristics (caTissue_Core_1_2 v1.2)
The combined anatomic state of a specimen

SpecimenProtocol (caTissue_Core_1_2 v1.2)
A set of procedures that govern the collection and/or distribution of biospecimens

SpecimenRequirement (caTissue_Core_1_2 v1.2)
Represents the required properties of a specimen while collecting and distributing it. SpecimenRequirement are...

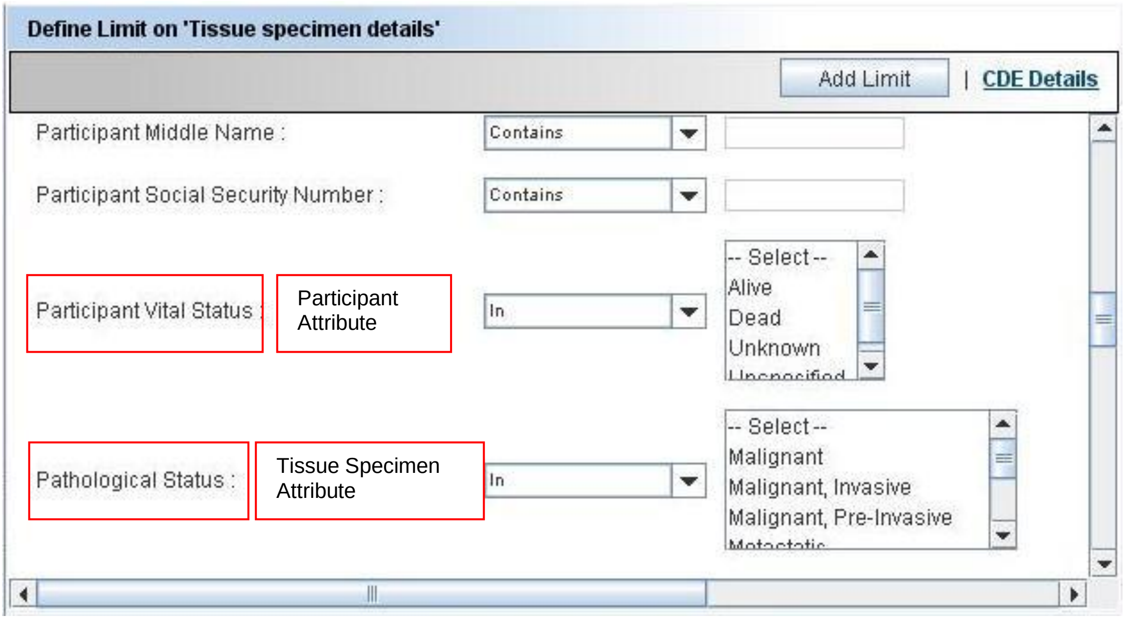
SpecimenArrayContent (caTissue_Core_1_2 v1.2)
An entity that defines the contents of a specimen array. It represents what all and how different specimens ar...

<< < 1 2 3 4 > >>

Next

Figure 5.14. Search category in caB2B Client Application

4. Click on the category name from the search results obtained. Application navigates to the **Add Limits** page.



Define Limit on 'Tissue specimen details'

Add Limit | CDE Details

Participant Middle Name : Contains

Participant Social Security Number : Contains

Participant Vital Status : In

-- Select --
Alive
Dead
Unknown
Unspecified

Participant Attribute

Pathological Status : In

-- Select --
Malignant
Malignant, Invasive
Malignant, Pre-Invasive
Metastatic

Tissue Specimen Attribute

Figure 5.15. Add Limits page for Category

5. Enter limit on the attributes.
6. Click **Add Limit** to add the node to DAG.
7. Click **Next** to execute the query (refer *caB2B Client Application End User manual to Execute query*)

All the queries available in the Web Application are created using categories. Thus, a web application user can see attributes from different entities together in the Define Limits view as well as the Results view.

Advantages of category

1. Creating a single category saves the time of adding all the classes required for the query.
2. User can view the results of all the classes in category together rather than navigating through related data.
3. User can add limits on one category in a single add limit page rather than adding to different classes.
4. User can add limits and query category without bothering about the path between the classes in a category.

Chapter 6 Curate Paths

This chapter describes how to curate paths. It contains the following topics:

- *Introduction to the Curated Paths*
- *Steps to Curate* Paths
- *Use of the Curated Path*

Introduction to the Curated Paths

As explained in the chapter *Load Models* when a model is loaded in caB2B MDR, application also stores all possible paths between each pair of classes. Due to this, when you or the user tries to connect two classes, application suggests all possible paths as explained in section *What is an Ambiguity Resolver?* In certain cases, ambiguity resolver might show more than hundred paths though only few of them are useful for queries. Using Curate Paths feature, you can curate the most useful and frequently used paths as the default paths. The curated paths reduce the caB2B users' task of selecting the most suitable and logical path from set of available paths.

For example, in case of caTissueCore v1.2 there are multiple paths between *SpecimenCollectionGroup* and *Specimen*. Most of these paths traverse through one of the event parameter classes, like *TransferEventParameter or FixedEventParameter*. A user may not be interested to know if the specimen has been thawed, fixed, or embedded. Most of the times user is interested to know what kind of specimens are available. To get such specimens, user will connect *Specimen* class directly to the *SpecimenCollectionGroup*. In such case, you can curate this path, so that application will readily show this path. It will save user's time to traverse through long list of all possible paths to find the path of interest.

You can also define a curated path to connect two or more classes. All the classes used to define a curated path should be from the same domain model. However, you cannot curate a path between the classes from different models like caFEv1.1 and GeneConnect.

Steps to Curate Paths

1. On the **Search Data** menu, click **Curate Paths**.

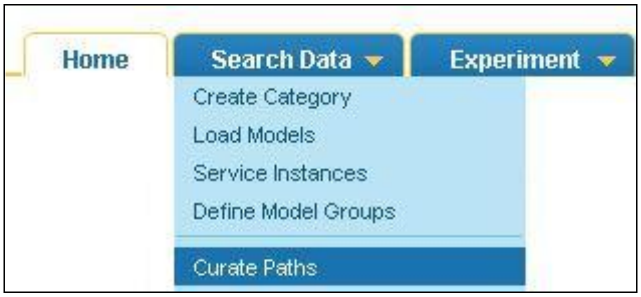
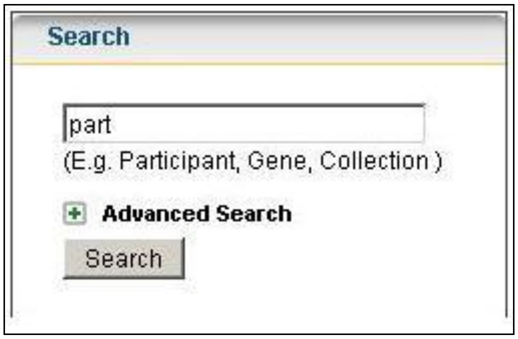


Figure 6.1. Search Data menu

2. Enter the class name in the search box.

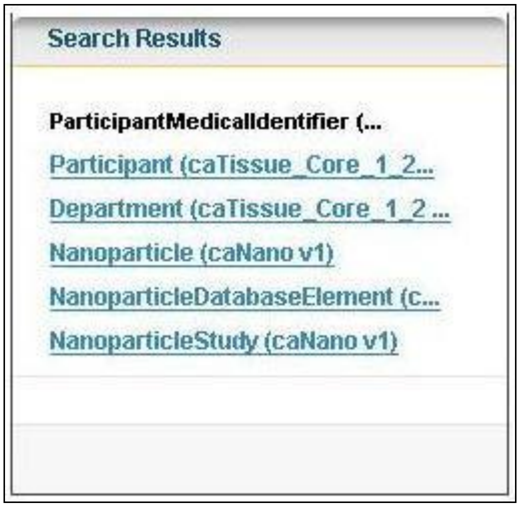


part

Figure 6.2. Text box to enter the search string

F.Y.I.	Choose the **Advanced Search** option to refine your search results. To know more about the advanced search options, refer to *Appendix A - Advanced Search.*

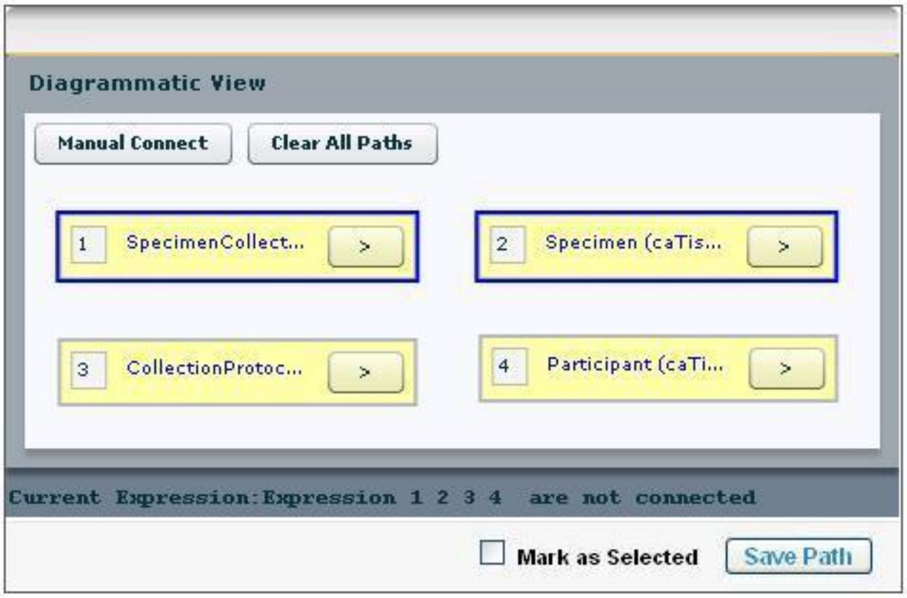
3. In the **Search Results**, click the class name to add it to the Diagrammatic View. Add two or more classes in the DAG.



ParticipantMedicalIdentifier (...
Participant (caTissue_Core_1_2...
Department (caTissue_Core_1_2 ...
Nanoparticle (caNano v1)
NanoparticleDatabaseElement (c...
NanoparticleStudy (caNano v1)

Figure 6.3. Select Class to add in DAG

4. In the DAG, select a pair of classes and click **Manual Connect**.



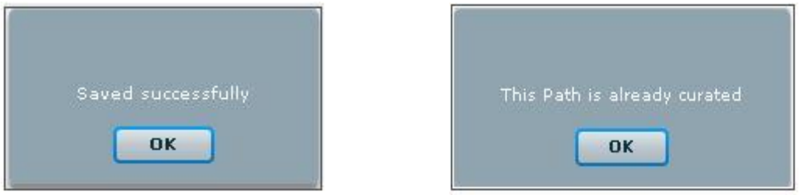
Current Expression:Expression 1 2 3 4 are not connected

Figure 6.4. Defining path between a pair of classes

5. Repeat above step to connect all the classes present in the DAG. To know more about how to connect the classes present in the DAG, refer to the section *Category Creation Rules.*
6. To force users to use a particular path, you can mark it as selected by checking **Mark as Selected** check box. When a path is marked as selected for a set of classes, whenever user tries to connect those classes, application will apply that path directly even though there are other available paths.

F.Y.I.	For a particular set of classes, you should only mark one path as selected. If you mark more than one as selected, application will apply the first one by default.

7. Click **Save Path**. A success message appears when the path is saved. If the path already exists, a warning message appears.



Saved successfully

This Path is already curated

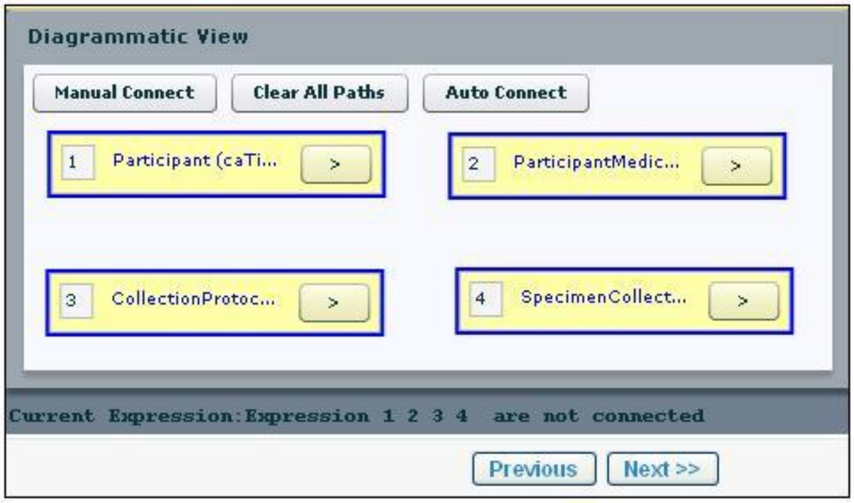
Figure 6.5. Messages

Use of the Curated Path

Use in Administrative Module

You can use a curated path for defining a category. To create categories, refer to the section *Steps to Create a Category*. Follow the steps below to use the curated path in a category:

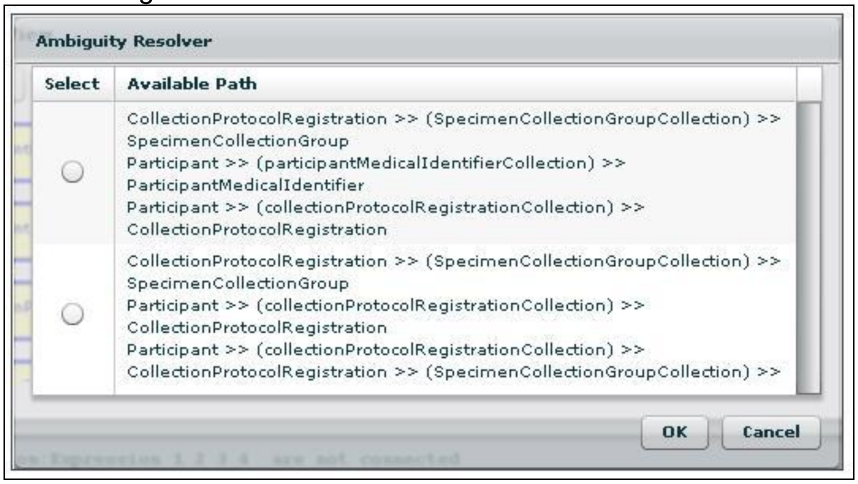
1. After adding the classes in DAG, select all the classes you wish to connect.



Current Expression:Expression 1 2 3 4 are not connected

Figure 6.6. Select all nodes

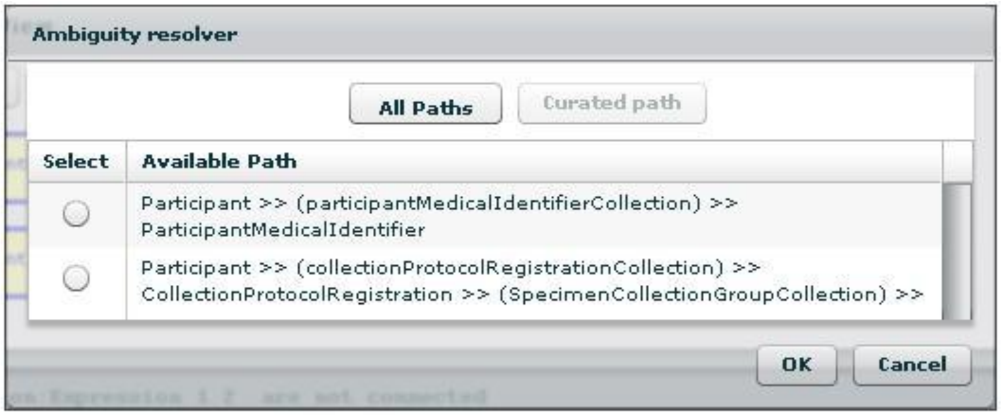
2. Click **Auto Connect** to connect all the classes with the defined curated path.
   a. If there is a single curated path for selected classes then, the application will select it by default.
   b. If multiple curated paths exist and none of them is marked as *selected*, then the Ambiguity Resolver window appears and you need to select the required path.
   c. If there is no curated path defined for selected classes, you will see a message as **No Paths Found**



Select	Available Path
○	CollectionProtocolRegistration >> (SpecimenCollectionGroupCollection) >> SpecimenCollectionGroup Participant >> (participantMedicalIdentifierCollection) >> ParticipantMedicalIdentifier Participant >> (collectionProtocolRegistrationCollection) >> CollectionProtocolRegistration
○	CollectionProtocolRegistration >> (SpecimenCollectionGroupCollection) >> SpecimenCollectionGroup Participant >> (collectionProtocolRegistrationCollection) >> CollectionProtocolRegistration Participant >> (collectionProtocolRegistrationCollection) >> CollectionProtocolRegistration >> (SpecimenCollectionGroupCollection) >>

Figure 6.7. Ambiguity Resolver window for multiple curated paths

3. In case you want to use path other than the curated paths, click **Manual Connect.** An Ambiguity Resolver window appears showing curated paths by default, click **All Paths** to see other paths, select desired path and click **OK.**



Select	Available Path
○	Participant >> (participantMedicalIdentifierCollection) >> ParticipantMedicalIdentifier
○	Participant >> (collectionProtocolRegistrationCollection) >> CollectionProtocolRegistration >> (SpecimenCollectionGroupCollection) >>

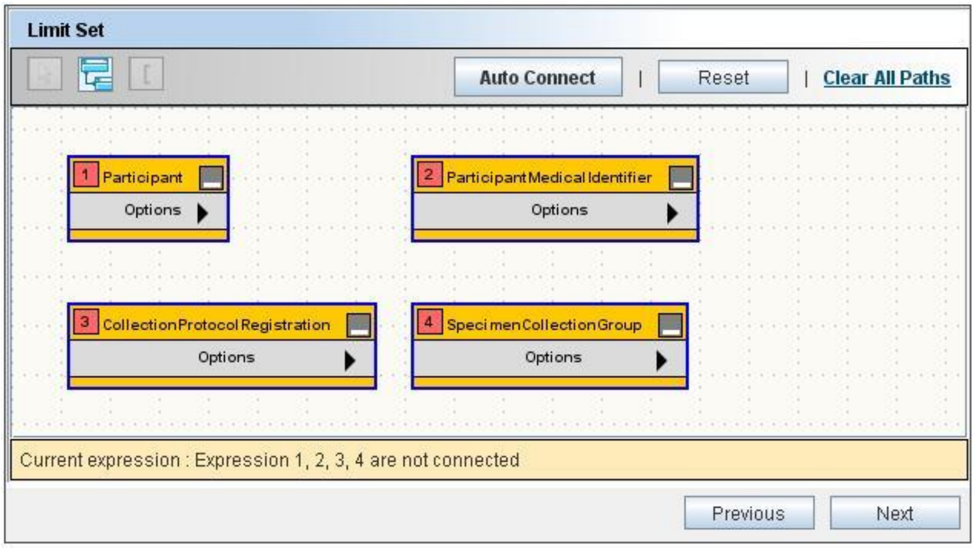
Figure 6.8. Curate path as default view

F.Y.I.	Manual connect works when you select only two classes in the Diagrammatic View

Use in Client Application

Client Application users can use the curated path while defining a query. Follow the steps below to define a query with the curated path:

1. Launch the Client Application and specify the limits (Refer to section Define Limits in *caB2B Client Application End User Manual*).
2. Select all the classes in DAG and click **Auto Connect** to connect the classes with the curated path.



Current expression : Expression 1, 2, 3, 4 are not connected

Figure 6.9. Selected nodes in caB2B Client Application

3. If you have marked the path as selected while curating the path, then the application uses that path even though you might have defined multiple curated paths.

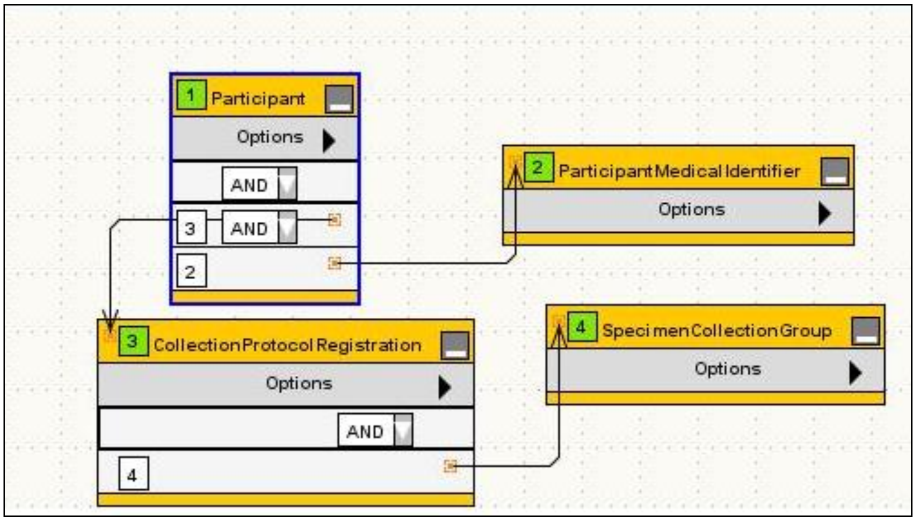
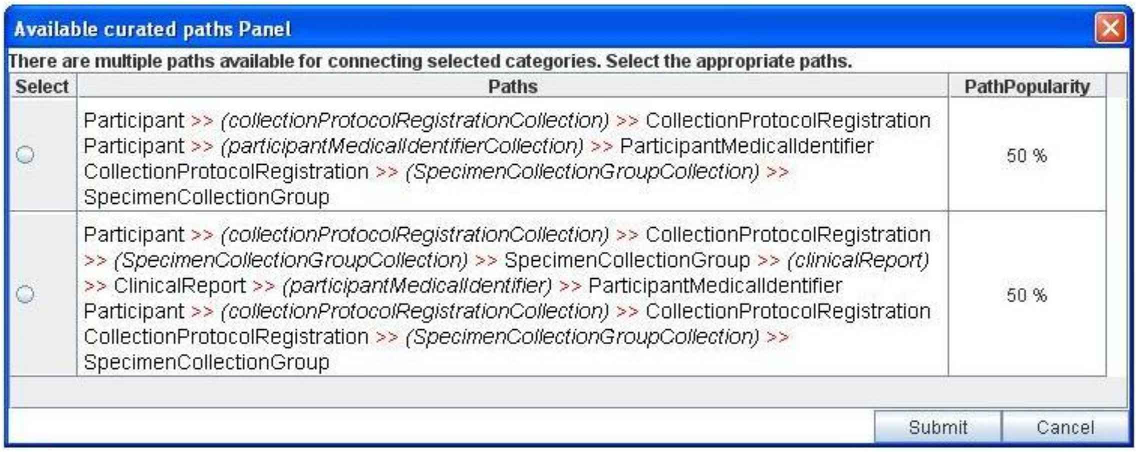

1 Participant
Options
AND
3 AND
2
2 ParticipantMedicalIdentifier
Options
3 CollectionProtocolRegistration
Options
AND
4
4 SpecimenCollectionGroup
Options

Figure 6.10. Classes connected in DAG during query using curated paths

4. If multiple curated paths exist then an ambiguity resolver window appears you can select the appropriate path to connect the classes.



Available curated paths Panel

There are multiple paths available for connecting selected categories. Select the appropriate paths.

Select	Paths	PathPopularity
○	Participant >> *(collectionProtocolRegistrationCollection)* >> CollectionProtocolRegistration Participant >> *(participantMedicalIdentifierCollection)* >> ParticipantMedicalIdentifier CollectionProtocolRegistration >> *(SpecimenCollectionGroupCollection)* >> SpecimenCollectionGroup	50 %
○	Participant >> *(collectionProtocolRegistrationCollection)* >> CollectionProtocolRegistration >> *(SpecimenCollectionGroupCollection)* >> SpecimenCollectionGroup >> *(clinicalReport)* >> ClinicalReport >> *(participantMedicalIdentifier)* >> ParticipantMedicalIdentifier Participant >> *(collectionProtocolRegistrationCollection)* >> CollectionProtocolRegistration CollectionProtocolRegistration >> *(SpecimenCollectionGroupCollection)* >> SpecimenCollectionGroup	50 %

Figure 6.11. Ambiguity resolver for more than one curated paths

Chapter 7 Inter Model Joins

This chapter describes how to create inter model joins. It contains the following topics:

- *Need for Inter Model Joins*
- *Introduction to Inter Model Joins*
- *Steps to define an Inter Model* Join
- *Using Inter model Join*

Need for Inter Model Joins

Many caGrid applications have data that is related. Like, caFE stores annotations of a particular gene whereas GeneConnect stores other information like mappings of the same gene. Unless you define the joins that depict the relationship between two applications, user will not be able to retrieve the data in single query.

For example, a translational scientist collects a sample and stores its information in a tissue banking software, such as caTissue. Later s/he performs a microarray experiment on the sample and stores this experiment data in the caArray repository. In future, when s/he wants to know the details of the specimen used in a microarray experiment, then s/he has to query the caTissue and caArray applications individually to retrieve the information. If there is a connection defined between caTissue and caArray model based specimen then s/he can directly retrieve the desired information with a single query.

Introduction to Inter Model Joins

Defining an Inter model join is similar to defining a path between classes from two different models. While creating path across two models, you can connect the classes, based on the following:

1. Public IDs: Attributes having same public ID
2. Concept Codes: Attributes having the same concept codes
3. Manual Settings: Manually define the common attribute

To know more about the public ID (Data Element ID) refer to, Common Data Element (CDE) browser (http://cdebrowser.nci.nih.gov/CDEBrowser/). To know more about the concept codes refer to NCI Metathesaurus (http://bioportal.nci.nih.gov).

Steps to define an Inter Model Join

1. On the **Search Data** menu, click **Create InterModel Join**.

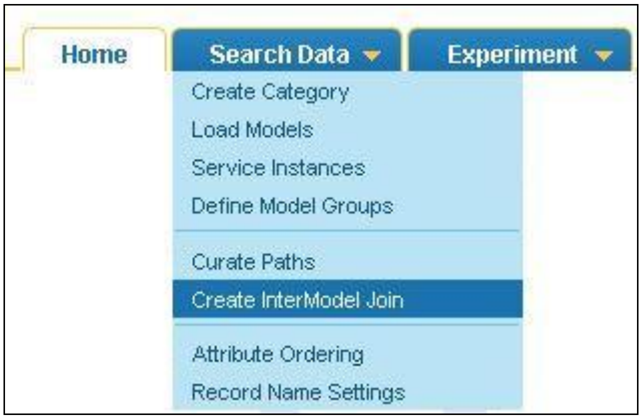


Figure 7.1. Search Data menu

2. Enter the class name in the search box and click the class name in the **Search Results**.

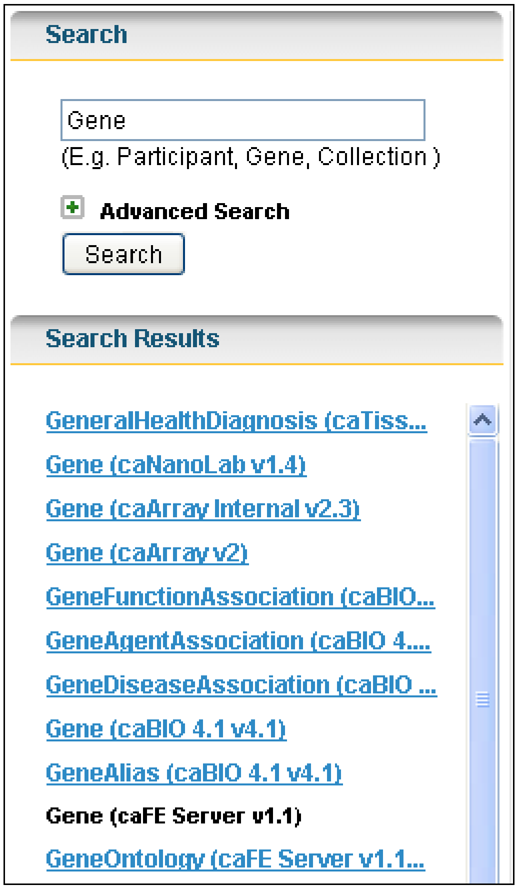
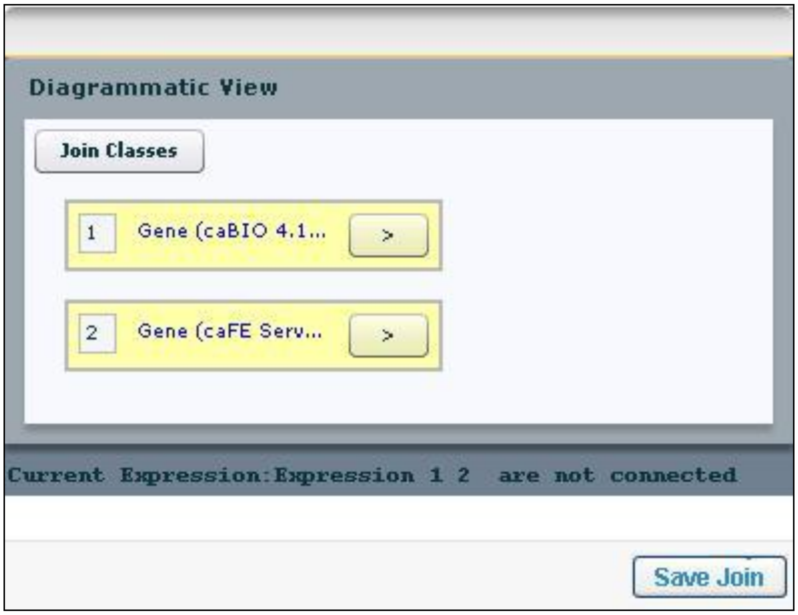

Search
Gene
(E.g. Participant, Gene, Collection)
Advanced Search
Search
Search Results
GeneralHealthDiagnosis (caTiss...
Gene (caNanoLab v1.4)
Gene (caArray Internal v2.3)
Gene (caArray v2)
GeneFunctionAssociation (caBIO...
GeneAgentAssociation (caBIO 4....
GeneDiseaseAssociation (caBIO ...
Gene (caBIO 4.1 v4.1)
GeneAlias (caBIO 4.1 v4.1)
Gene (caFE Server v1.1)
GeneOntology (caFE Server v1.1...

Figure 7.2. Search

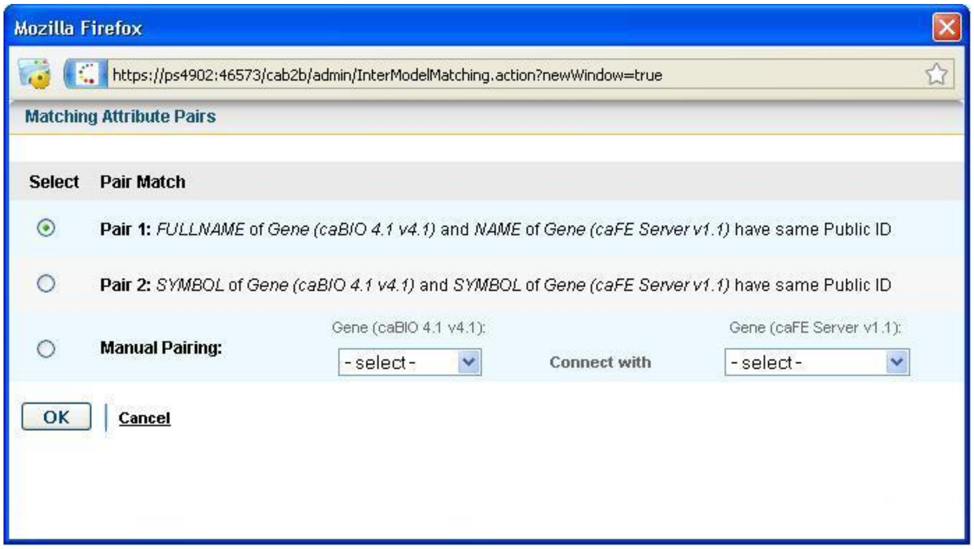
3. The class appears in the DAG as shown below. Now add the second class to the DAG.



Current Expression:Expression 1 2 are not connected

Figure 7.3. DAG Panel

4. Select both the classes and then click on **Join Classes**. A pop-up window appears. From the pop-up window, select one of the available paths.


Mozilla Firefox
https://ps4902:46573/cab2b/admin/InterModelMatching.action?newWindow=true
Matching Attribute Pairs
Select
Pair Match
OK
Cancel

● **Pair 1:** *FULLNAME* of *Gene (caBIO 4.1 v4.1)* and *NAME* of *Gene (caFE Server v1.1)* have same Public ID

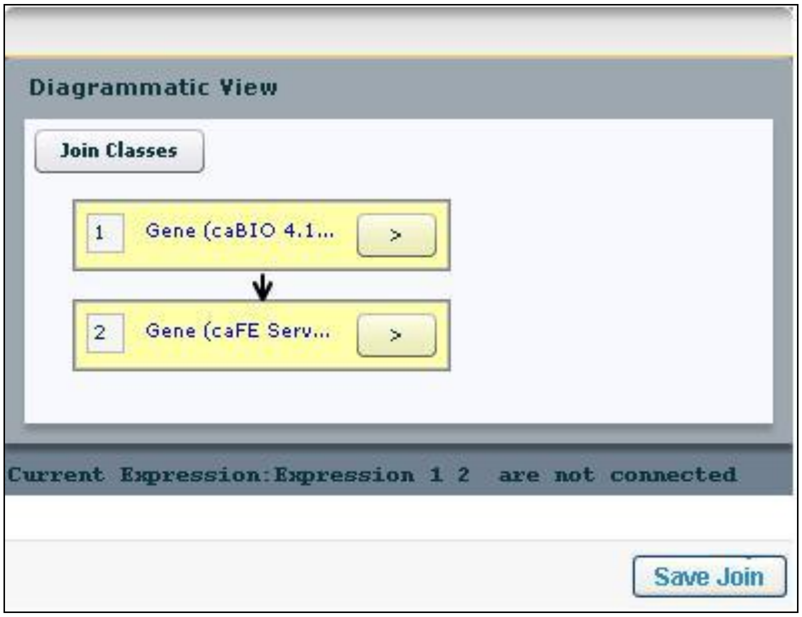
○ **Pair 2:** *SYMBOL* of *Gene (caBIO 4.1 v4.1)* and *SYMBOL* of *Gene (caFE Server v1.1)* have same Public ID

○ **Manual Pairing:** Gene (caBIO 4.1 v4.1): - select - Connect with Gene (caFE Server v1.1): - select -

Figure 7.4. Pop-up window displaying the paths

5. If no matches are available or you want to define a join a manually select the attributes from the option **Manual Pairing**

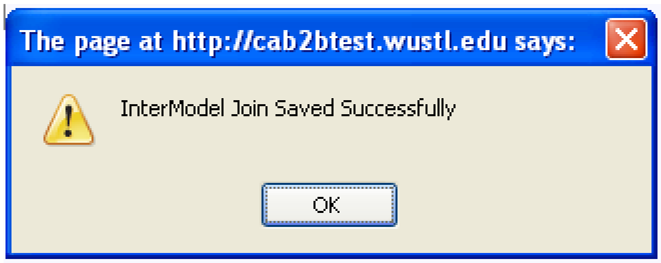
6. Click **OK** to save the join.



Current Expression:Expression 1 2 are not connected

Figure 7.5. Classes connected in DAG

7. Once the nodes are connected, click **Save Path**. A success message appears if the inter model join is saved.

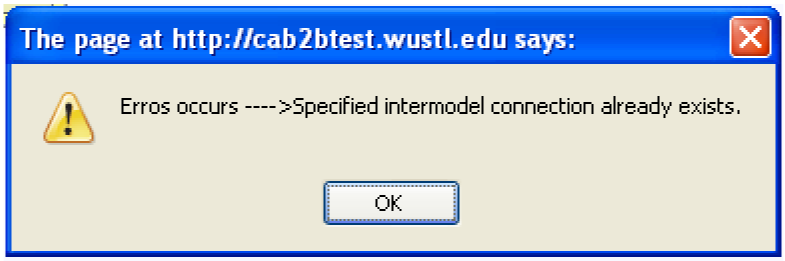


The page at http://cab2btest.wustl.edu says:

InterModel Join Saved Successfully

Figure 7.6. Success Message

8. An error message appears if the path already exists.



The page at http://cab2btest.wustl.edu says:

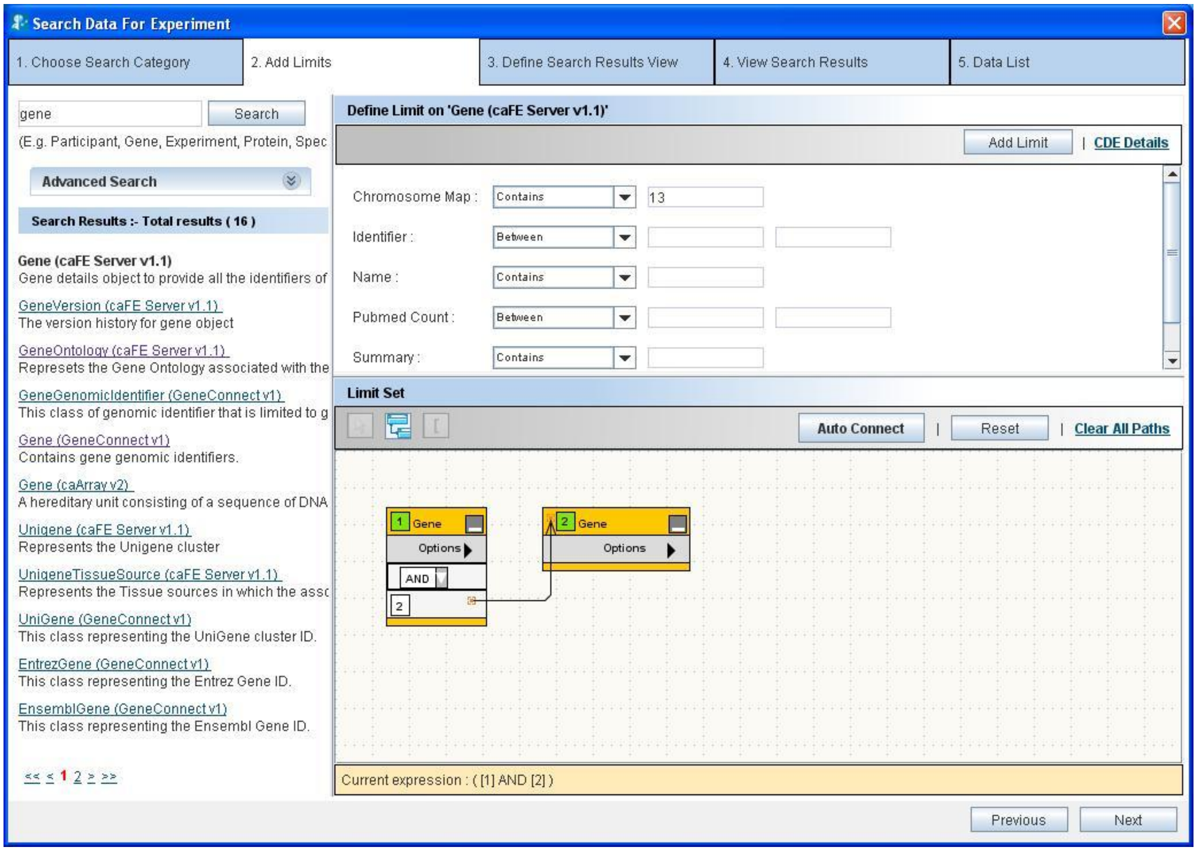
Erros occurs ---->Specified intermodel connection already exists.

Figure 7.7. Failure Message

Using Inter model Join

Client Application users can use the inter model join while querying.

Example: If you create an inter model join between *Gene (caFE)* and *Gene (Geneconnect)*, you can use this path to query across caFE and Geneconnect.



gene

(E.g. Participant, Gene, Experiment, Protein, Spec

Search Results :- Total results (16)

Gene (caFE Server v1.1)
Gene details object to provide all the identifiers of

GeneVersion (caFE Server v1.1)
The version history for gene object

GeneOntology (caFE Server v1.1)
Represets the Gene Ontology associated with the

GeneGenomicIdentifier (GeneConnect v1)
This class of genomic identifier that is limited to g

Gene (GeneConnect v1)
Contains gene genomic identifiers.

Gene (caArray v2)
A hereditary unit consisting of a sequence of DNA

Unigene (caFE Server v1.1)
Represents the Unigene cluster

UnigeneTissueSource (caFE Server v1.1)
Represents the Tissue sources in which the assc

UniGene (GeneConnect v1)
This class representing the UniGene cluster ID.

EntrezGene (GeneConnect v1)
This class representing the Entrez Gene ID.

EnsemblGene (GeneConnect v1)
This class representing the Ensembl Gene ID.

Define Limit on 'Gene (caFE Server v1.1)'

Chromosome Map : Contains 13
Identifier : Between
Name : Contains
Pubmed Count : Between
Summary : Contains

Current expression : ([1] AND [2])

Figure 7.8. Classes connected in DAG as per the inter model joins defined

Chapter 8 Multi-model Category

This chapter describes the details of multi-model category. It contains the following topics:

- *Introduction to Multi-model Category*
- *Creating Multi-model Category*
- *Using Multi-model Categories*

Introduction to Multi-model Category

Multi model category (MMC) is single representation of data that is stored in different applications. For example, biospecimen data can be stored in different tissue banking repositories like, caTissue Suite or caTissue Core.

Why we need Multi model category?

caGrid offers multiple applications to store particular kind of data. Current caGrid infrastructure does not provide a way to execute single query on all such applications. However, it does support querying individual applications using their metadata. Building such queries requires knowledge of each application. In addition, due to the differences in the metadata of these applications, a single query cannot fetch data from all applications; user has to execute one query for each application.

User should be able to query the data irrespective of the application used to store it. MMC provide user a way to query across such multiple applications. A MMC consist of classes from multiple models; these may be different versions of same application or different applications. The attributes from different models having the similar data correspond to a single attribute in the MMC

For example, caGrid offers multiple tissue banking repository applications like caTissue Core and caTissue Suite. As they have different model, a single query cannot fetch data from both. You can create a MMC using these two models by mapping the attributes, like the attribute *Specimen Type* from both the applications should correspond to a single attribute in the MMC. You can use this MMC to build query to fetch data from both the applications together. Using this query, user can find out available bio-specimens for *Breast Adenocarcinoma* from both tissue banking applications.

Creating Multi-model Category

In this caB2B release, you have to create MMC using command line utilities. User interface is not available for creating them.

Environment Setup for command line utilities

To use command line utilities, you need the caB2B installer zip, refer to **caB2B Installation manual** to get caB2B installer. Unzip the installer in to a folder, say `C:/cab2b` for Windows or `/home/cab2b` for Linux (hereafter referred as `cab2b_home`). You need to modify a few properties in a configuration file `{cab2b_home}/software/cab2b/src/conf/hibernate.cfg.xml`

Open this file in any text editor program and replace `<property name="connection.datasource">java:/DS-NAME</property>` by the contents present in the box below:

```
<!-- property name="connection.datasource">java:/DS-NAME</property -->
<property name="hibernate.connection.driver_class"
>com.mysql.jdbc.Driver</property>
<property name="hibernate.connection.username">db_user</property>
<property name="hibernate.connection.password">db_password</property>
<property name="hibernate.connection.url"
>jdbc:mysql://db_server_host:db_server_port/database_name</property>
```

Now do following changes.

1. Replace `database_name` with name of the database schema used by your caB2B server
2. Replace `db_user` with the user name for the above database.
3. Replace `db_password` with the password of the above database user.
4. Replace `db_server_host` with the hostname of the MySQL server on which above database is located.

F.Y.I.	In case your installer and the MySQL server are on different machines, ensure that above mentioned user is allowed to access the mentioned database from the machine on which you are running installer

5. Replace `db_server_port` with the port on which, above MySQL server is running.

F.Y.I.	In case you are not sure about any of the above properties, please contact your caB2B installation owner. If you are the owner, you can find these details in file `JBOSS_HOME/server/{SERVER_NAME}/deploy/cab2b-ds.xml` here `{SERVER_NAME}` is Jboss configuration used for caB2B.

Working with MMC XML

For creating MMC, you need to define its attribute and their mapping in each of the individual models. You also need to specify the paths that connect the classes in the models. Creation of a multi-model category involves defining a XML file. You need some preliminary knowledge of XML editing for this. Refer to http://www.w3schools.com/xml/default.asp. The XML file should have the following structure:

```
<MultiModelCategory name=" " description=" " applicationGroup=" ">
<MultiModelCategoryAttributes>
     <MultiModelCategoryAttribute name=" " description=" ">
     <Attribute name=" " model=" " class=" " />
     <Attribute name=" " model=" " class=" " />
</MultiModelCategoryAttribute>
<Paths>
     <Path id="" />
     <Path id="" />
</Paths>
```

Details of this XML are provided below:

1. `<MultiModelCategory>` this is the root element. One file should contain only one `<MultiModelCategory>` element.

   Specify the `name-description` for the MMC. Each MMC belongs to one model group (refer to chapter *Define Model Groups).* Specify name of the model group to which this MMC belongs in `applicationGroup`

2. One MMC will have many attributes. Each such attribute is called Multi-model attribute (MMA). For each MMA create one tag `<MultiModelCategoryAttribute`>. Specify the `name-description` for the MMA

3. One MMA maps to one or more individual model attributes. One `<Attribute>` represents one attribute from a model. If you want to map MMA to two model-attributes, then you will have two `<Attribute>.` Specify `name`, `model`, and `class` for each model attribute.

4. Individual model attributes can come from different classes

5. `<Paths>` depicts the ways in which classes present in MMC and belonging to same model should be connected. `<Path id="" />` represents identifier of one such path (refer to section: *Introduction to Path)*

   While model loading, application also generates these paths. To know more about loading models refer to *Chapter 2 - Load Models*. These paths define the associations present between the classes. You need to use another command line utility to find the path identifiers. The XML file structure to get path identifiers is as follows:

```
<pairs>
  <pair>
   <source>edu.wustl.catissue.domain.Participant</source>
   <target>edu.wustl.catissue.domain.MolecularSpecimen</target>
  </pair>
  <pair>…</pair>
  .
  .
</pairs>
```

   These XML elements represent:

   - `Source`: Specify the fully qualified name for the source class, e.g., `edu.wustl.catissuecore.domain.Participant`.

   - `Target`: Specify the fully qualified name of the target class, e.g., `edu.wustl.catissuecore.domain.MolecularSpecimen`.

You need to specify all such pairs that are present in the category. To know more about how to specify the paths to connect the categories refer to, *Category Creation Rules.* To generate the path identifiers run the following commands. Open the shell-

Windows: **Start->All Programs-> Accessories-> Command Prompt**
Linux: **Main Menu-> System Tools-> Terminal** To generate the path ids:

a) `cd <cab2b_home>/software/cab2b`

b) `ant find.path.ids`

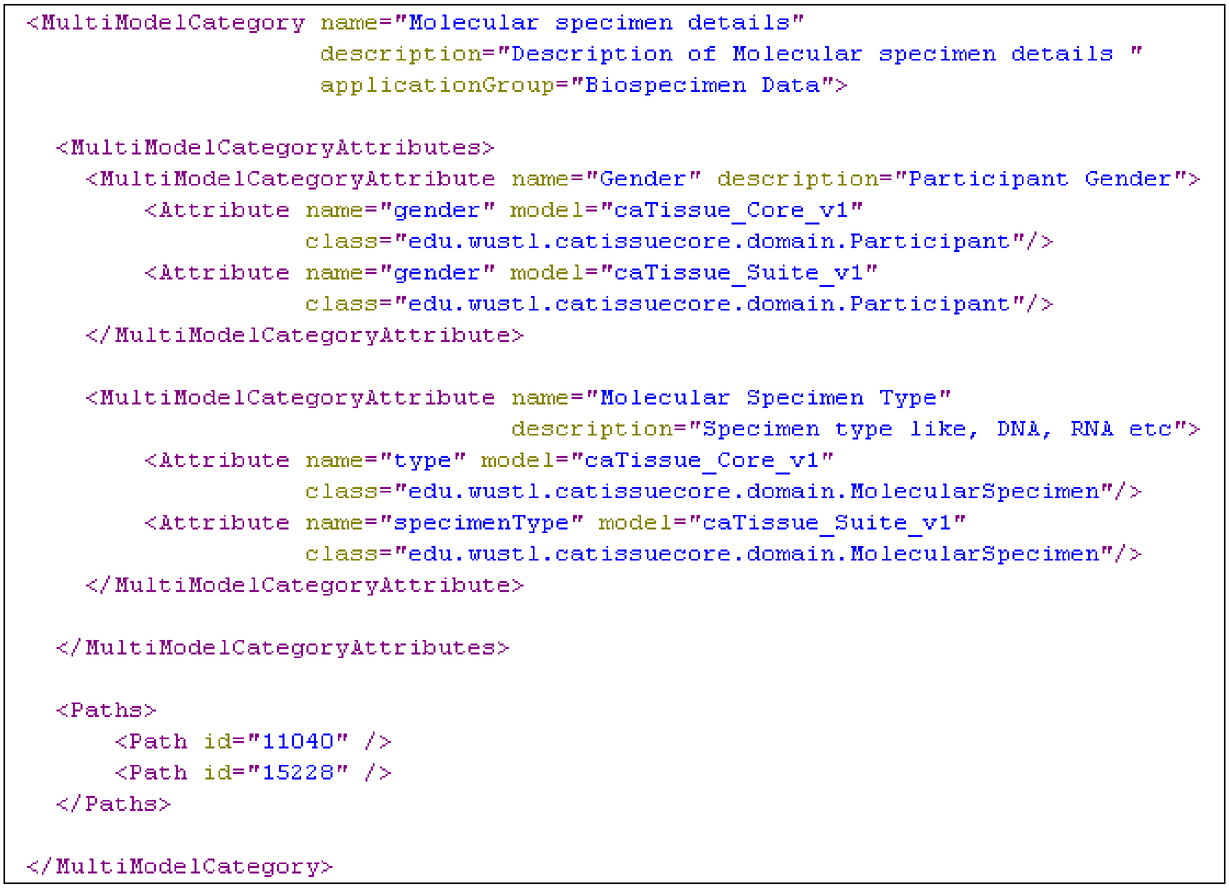
c) Provide the path for the XML file.

d) Provide the path for the output file.

After execution, this utility generates a file with all the paths present between the given pairs. From this set, find the path that is required in the MMC and add its identifier to the MMC XML. You need to specify the paths between all the pair of classes from the models present in the category.

Example MMC XML

Following is an example of the multi-model category XML



```
<MultiModelCategory name="Molecular specimen details"
                    description="Description of Molecular specimen details "
                    applicationGroup="Biospecimen Data">

  <MultiModelCategoryAttributes>
    <MultiModelCategoryAttribute name="Gender" description="Participant Gender">
        <Attribute name="gender" model="caTissue_Core_v1"
                   class="edu.wustl.catissuecore.domain.Participant"/>
        <Attribute name="gender" model="caTissue_Suite_v1"
                   class="edu.wustl.catissuecore.domain.Participant"/>
    </MultiModelCategoryAttribute>

    <MultiModelCategoryAttribute name="Molecular Specimen Type"
                                 description="Specimen type like, DNA, RNA etc">
        <Attribute name="type" model="caTissue_Core_v1"
                   class="edu.wustl.catissuecore.domain.MolecularSpecimen"/>
        <Attribute name="specimenType" model="caTissue_Suite_v1"
                   class="edu.wustl.catissuecore.domain.MolecularSpecimen"/>
    </MultiModelCategoryAttribute>

  </MultiModelCategoryAttributes>

  <Paths>
      <Path id="11040" />
      <Path id="15228" />
  </Paths>

</MultiModelCategory>
```

Figure 8.1. Example MMC XML

This XML will create a MMC *Molecular specimen details* belonging to *Biospecimen data* model group. It would have two attributes *Gender* and *Molecular Specimen Type*.

- The *Gender* attribute corresponds to *gender* attribute from the *Participant* class from *caTissuecore v1.2* and *caTissueSuite v1.1*.
- The *Molecular Specimen Type* attribute refers to *type* attribute from the *Molecular Specimen* class from *caTissuecore v1.2* and *specimentype* attribute from the *Molecular Specimen* class from *caTissueSuite v1.1*.
- The *Path id* is the identifier for the path that connects *Participant* to *Molecular Specimen* in *caTissuecore v1.2* and *caTissueSuite v1.1.*

Loading MMC XML

After defining all the attributes, classes, paths etc you can use this XML to create the MMC. Execute the following commands:

Open the shell-
Windows: **Start->All Programs-> Accessories-> Command Prompt**
Linux: **Main Menu-> System Tools-> Terminal**
To create the categories:

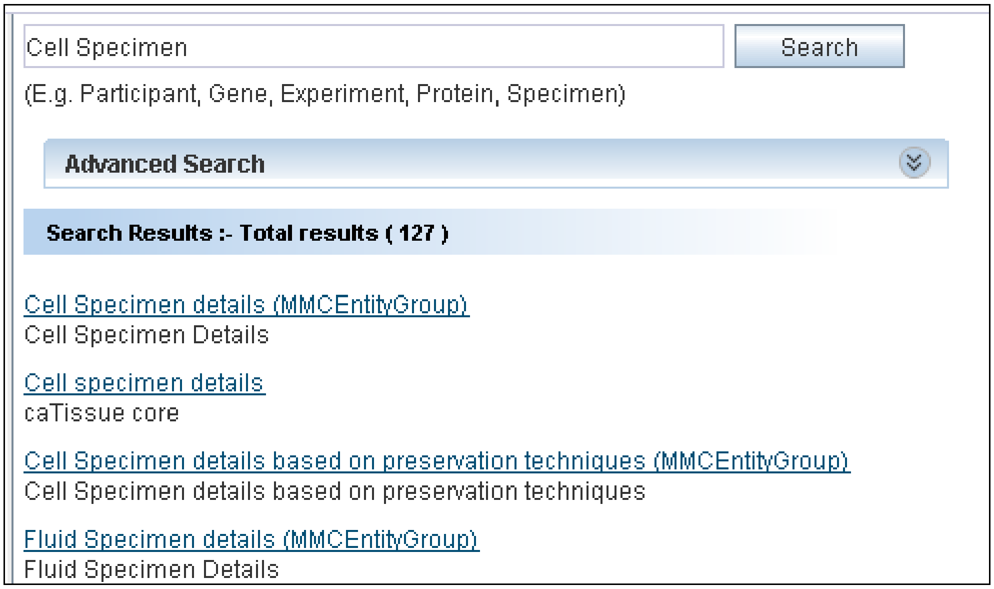
```
a. cd <cab2b_home>/software/cab2b

b. ant load.mmcs
```

This would create the MMC. You can launch the Client Application and search for the created categories using the metadata search.

Using Multi-model Categories

MMCs are available in the Client Application and the Web Application. You can use MMC to create a query in the Client Application. These queries are available in the Web Application. A Web Application user can execute the multi-model queries to retrieve data from multiple instances at the same time.

For example, you want to find data from biospecimen repository located at Washington University and Indiana University. Washington University stores its data using caTissuecore v1.2 and Indiana University uses caTissueSuite v1.1. You can create a MMC with the metadata from the caTissuecore v1.2 and caTissueSuite v1.1. Create a query using this MMC to query both the instances of caTissue. After creating the MMC, you can search for it in the Client Application. Figure below shows the metadata search results for the MMC:



(E.g. Participant, Gene, Experiment, Protein, Specimen)

Search Results :- Total results (127)

Cell Specimen details (MMCEntityGroup)
Cell Specimen Details

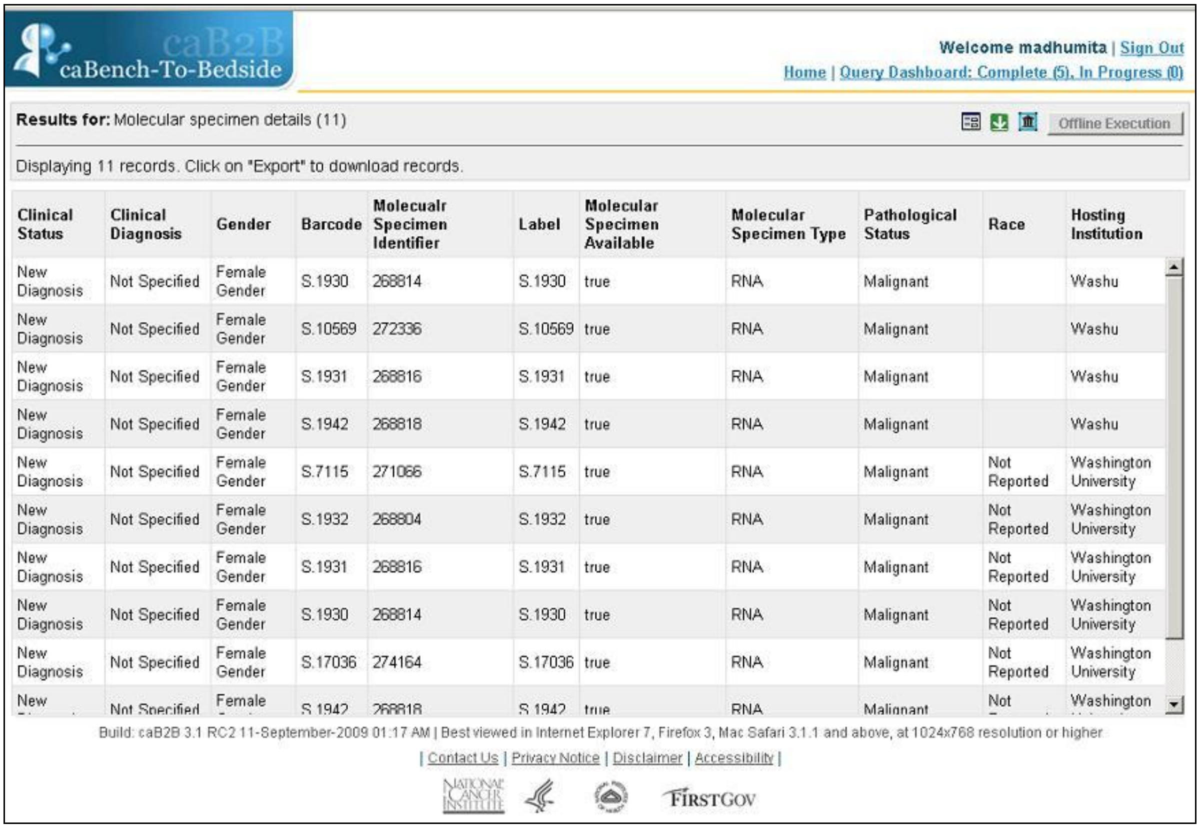
Cell specimen details
caTissue core

Cell Specimen details based on preservation techniques (MMCEntityGroup)
Cell Specimen details based on preservation techniques

Fluid Specimen details (MMCEntityGroup)
Fluid Specimen Details

Figure 8.2. Metadata search in Client Application

To execute the queries created using this category use the Web Application. Execute a query created using the MMC using one service of caTissuecore v1.2 and one service for caTissueSuite v1.1. You would retrieve results from both the services.



caB2B
caBench-To-Bedside
Welcome madhumita | Sign Out
Home | Query Dashboard: Complete (5), In Progress (0)
Offline Execution

Results for: Molecular specimen details (11)

Displaying 11 records. Click on "Export" to download records.

Clinical Status	Clinical Diagnosis	Gender	Barcode	Molecualr Specimen Identifier	Label	Molecular Specimen Available	Molecular Specimen Type	Pathological Status	Race	Hosting Institution
New Diagnosis	Not Specified	Female Gender	S.1930	268814	S.1930	true	RNA	Malignant		Washu
New Diagnosis	Not Specified	Female Gender	S.10569	272336	S.10569	true	RNA	Malignant		Washu
New Diagnosis	Not Specified	Female Gender	S.1931	268816	S.1931	true	RNA	Malignant		Washu
New Diagnosis	Not Specified	Female Gender	S.1942	268818	S.1942	true	RNA	Malignant		Washu
New Diagnosis	Not Specified	Female Gender	S.7115	271066	S.7115	true	RNA	Malignant	Not Reported	Washington University
New Diagnosis	Not Specified	Female Gender	S.1932	268804	S.1932	true	RNA	Malignant	Not Reported	Washington University
New Diagnosis	Not Specified	Female Gender	S.1931	268816	S.1931	true	RNA	Malignant	Not Reported	Washington University
New Diagnosis	Not Specified	Female Gender	S.1930	268814	S.1930	true	RNA	Malignant	Not Reported	Washington University
New Diagnosis	Not Specified	Female Gender	S.17036	274164	S.17036	true	RNA	Malignant	Not Reported	Washington University
New	Not Specified	Female	S.1942	268818	S.1942	true	RNA	Malignant	Not	Washington

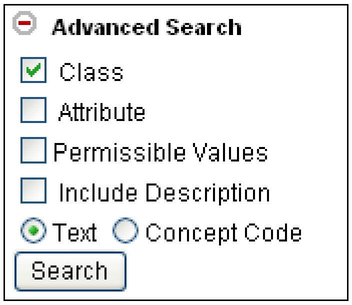
Build: caB2B 3.1 RC2 11-September-2009 01:17 AM | Best viewed in Internet Explorer 7, Firefox 3, Mac Safari 3.1.1 and above, at 1024x768 resolution or higher

| Contact Us | Privacy Notice | Disclaimer | Accessibility |

Figure 8.3. Results of MMC query

Appendix A Advanced Search

In case application returns too many results for your class search, you can use the **Advanced Search** to narrow your search. Expanding the **Advanced Search** box displays advanced options that help you refine the category search.



Advanced Search

☑ Class

□ Attribute

□ Permissible Values

□ Include Description

● Text ○ Concept Code

Figure A.1. Advanced search

Table A.1. Advanced Search Options (Checkboxes and radio buttons)

Option	***Description***
Check Boxes	
Category	Includes the category names in the search
Attribute	A category contains many attributes. Select this check box to include the individual attribute names in the search. For example, Select *attribute* and search for *gender*. You will find the category Participant since it contains an attribute by name *Gender*.
Permissible value	Some attributes can have fixed set of values. For example, *Gender* can be *Male, Female or Unspecified. Organ* can be *heart, lung, breast,* and so on. **Permissible Values** is a set of all such values. Select this check box to include the permissible values in the search. For example, if you select the permissible value as *Male* the system displays the Participant category since it contains an attribute *Gender* that contains a permissible value *Male gender*.
Include Description	Includes the description text in the search **Note**: **Include Description** check-box is enabled only for a **text** based search when you select the **Category** check box**,** the **Attribute** check box, or both.
Option Buttons	
Text	The system matches the search string that you type with the name of the category, its attributes, or the permissible values based on the text. This is simple, most commonly used and the default way of searching..
Concept Code	The system treats the search string as an **E**nterprise **V**ocabulary **S**ervices (EVS) concept code and matches it with the concept code of the category, its attributes, or the permissible values thereby showing the matching categories. More details on the concept code can be found at http://bioportal.nci.nih.gov

Index